

分类号.....

密级.....

UDC.....

编号.....

中南大学

CENTRAL SOUTH UNIVERSITY

# 硕士学位论文

论 文 题 目 基于 C/S 结构的教育储蓄个人  
所得税管理系统的设计与开发

学 科、专 业 计算机技术

研 究 生 姓 名 吴芙蓉

导 师 姓 名 及  
专 业 技 术 职 务 桂卫华 教授

## 摘 要

论文介绍了基于 C/S(Client/Server)结构的教育储蓄个人所得税管理系统设计与开发。所开发的教育储蓄个人所得税管理系统用于国税系统对利息个人所得税中教育储蓄的日常征收管理。该系统遵循经济、高效、可靠、易操作、易维护和开放的设计原则,采用 Client/Server 计算模式,以 Windows2000 Server 为网络操作系统,以 MS SQL Server2000 为后台数据库系统,采用 Powerbuilder9.0 作为客户端应用程序开发工具。实现了日常管理、数据采集、数据比对、违规处理、查询统计、系统维护等功能。

论文在系统需求分析的基础上,对系统的功能模块、拓扑结构、数据库设计等进行了较详细的论述。针对教育储蓄个人所得税管理系统在国税征管中的应用范围、应用对象的特点,确定了系统的基于 C/S 结构的开发模型和技术方案。并对存储过程、OLE DB 技术、安全设计、系统性能优化技术等在该系统中的具体实现进行了详细的阐述。

教育储蓄个人所得税管理系统用户界面友好,安装使用简单实用,在系统投入使用的试运行阶段取得了良好的成效,查出大量违规的银行、学校和储户,查补了大量利息个人所得税。运行结果表明该系统作为主体征管软件的外延拓展补充,满足了教育储蓄利息个人所得税征收管理的实际工作需要。

**关键字:** C/S 模型, 教育储蓄个人所得税管理系统, OLE DB, 性能优化

## ABSTRACT

This paper introduces the design and the development of ESMS(Education Saving Management System) which based on Client/Server Software Architectures. Follow the principles of economic, safety, efficiently, credibility, easy operation, easy maintenance and open design, Client/ Server Software Architecture is adopted, Windows 2000 Server as the network operate system is selected , choosing the MS SQL Server 2000 as the backstage database system is choosed, and Powerbuilder9.0 as client's program development tool is applied. In order to satisfy day-to-day tax celection management of indiviual income tax, ESMS carried out mainly as follows function: daily management; information collection; data comparison; irregularity disposal; inquiry and statistic; system maintenance etc.

On the basis of requirement analysis, it discusses detailly foundation module, topological structure and database design...etc. Acording to application range and feature in the state administration of taxation system, Client/Server Software Architectures and technique project suitable for system are adopted. This thesis also elaborates the realization of a system design in the key technique that included database access technique — OLE DB, storage procedure , safety design and system performance optimization technique in system development.

With the charactristics of friendly user interface, easy set-up and practical, ESMS have obtained good results and avoided a great deal of evasion of indiviual income taxes, dealt with a lot of irregularity bank, school and depositor, The result indicates that the system's design totally achieves the prospective aim, satisfied completely daily managemetn of interest indiviual income taxation.

**KEY WORDS:** Client/Server Software Architecture, Education Saving Management System, OLE DB, Performance optimization

# 目 录

摘 要.....	I
第一章 绪论.....	1
1.1 课题研究背景.....	1
1.2 国内外研究现状 .....	2
1.3 课题研究的目的是和意义 .....	4
1.4 论文的组织.....	5
第二章 系统需求分析 .....	6
2.1 教育储蓄个人所得税管理系统的总体目标.....	6
2.2 教育储蓄个人所得税管理业务要求.....	7
2.3 系统的基本业务流程.....	9
2.4 系统数据流程图 .....	10
2.5 本章小结 .....	12
第三章 系统总体设计 .....	13
3.1 系统的网络结构 .....	13
3.2 系统运行环境.....	14
3.3 软件功能描述.....	14
3.4 系统模块结构.....	15
3.5 本章小结 .....	18
第四章 系统关键技术及其实现 .....	19
4.1 开发模型 .....	19

4.2 数据库设计 .....	22
4.2.1 设计原则 .....	22
4.2.2 数据建模 .....	24
4.2.3 证明单流水号功能的实现 .....	26
4.3 信息编码技术 .....	29
4.3.1 信息编码技术的概述 .....	31
4.3.2 操作员编码的设计 .....	31
4.4 OLE DB 技术 .....	32
4.4.1 OLE DB 技术的概念 .....	32
4.4.2 Microsoft 数据访问组件 (MDAC) .....	34
4.4.3 OLE DB 在系统中的具体实现 .....	36
4.5 系统性能优化 .....	37
4.6 系统安全技术 .....	44
4.6.1 数据库安全技术 .....	45
4.6.2 基于应用系统的安全技术 .....	52
4.6.3 基于网络的安全技术 .....	53
4.7 本章小结 .....	53
<b>第五章 系统应用 .....</b>	<b>55</b>
5.1 系统的用户界面 .....	55
5.2 系统测试及应用 .....	59
5.3 本章小结 .....	61
<b>第六章 结论及展望 .....</b>	<b>62</b>
<b>参考文献 .....</b>	<b>64</b>
<b>致 谢 .....</b>	<b>67</b>
<b>硕士研究生期间发表论文 .....</b>	<b>68</b>

## 第一章 绪论

### 1.1 课题研究背景

随着政府信息化进程的加速,利用信息化手段提高税务行业的管理效率已经成为共识。国家税务总局早在 1996 年就提出了“以纳税申报和优化服务为基础,以计算机网络为依托,集中征收,重点稽查相结合”的税收征管新模式<sup>[1]</sup>。按照税务信息一体化建设的总体目标,金税一、二期的建设,建立了覆盖省、市、县广域网并与总局构成四级网络,基础设施建设初具规模。配合信息化建设的需要,综合征管软件——CTAIS(中国税收管理信息系统)在全国推广应用,计算机应用以“金税工程”为先导,逐步深入到税收征收管理、出口退税、办公自动化、财务、监察、反避税、数据分析、系统监控等各个领域。但利息所得个人所得税开征后,由于政策的变化性及业务的特殊性,没有将教育储蓄的管理纳入综合征管软件的功能模块中,而这一块也因此失去了税收信息化的科学管理。

1999 年恢复开征储蓄存款利息所得个人所得税(以下简称利息税)以后,根据科教兴国的战略,为促进国家教育事业健康发展,鼓励城乡居民以储蓄存款方式为其子女接受非义务教育积蓄资金,国家决定自 2000 年开始设立教育储蓄账户,给予利率及免征利息税的优惠。相对其他储蓄存款而言,教育储蓄有三方面好处:一是家庭可以为其子女(或被监护人)接受非义务教育(指九年义务教育之外的全日制高中、大中专、大学本科、硕士和博士研究生)在储蓄机构通过零存整取方式积蓄资金;二是符合规定的教育储蓄专户,可以享受整存整取利率的优惠;三是教育储蓄存款的利息免征个人所得税<sup>[2]</sup>。

该项制度出台以来,各有关部门和银行金融机构积极配合,使这项税收优惠政策得以落实,对一些困难家庭子女接受高等教育提供了帮助,在一定程度上给广大居民带来了实惠,对促进教育事业的发展起到了积极的作用。但是同时也存在以下的问题:

(1) 由于原教育储蓄管理办法对正在接受非义务教育学生身份证明的格式、印制、开具等没有做出统一规范,一些地区教育储蓄业务管理不严格、不规范,政策执行变形,造成国家税收流失。

(2) 由于教育储蓄存入时按零存整取、支取时按整存整取利率支付利息,同时又可以免征利息所得税,有的储户就想各种办法违规办理教育储蓄,如化整为零、超限额存款、多银行存款、多地域存款等方法,以逃避利息个人所得税的缴纳。

(3) 各金融机构激烈的竞争和存款余额的压力,各金融机构储蓄网点已将

教育储蓄作为揽储的一种重要手段，以致各金融机构为了增加存款，在办理和支取储蓄存款时，没有严格按照教育储蓄管理条例的规定进行操作，无论是否符合条件，都为储户开办教育储蓄。

(4) 即使有完善规范的征管政策和办法，但由于目前各金融机构之间信息不能共享，教育储蓄管理条例所规定的一个储户原则只能开具一个教育储蓄账户及账户本金不能超过 2 万元的规定在实际工作中无法落实，学校开具的非义务教育证明也有些失控，教育储蓄利息所得免征个人所得税已成为了利息所得税征管的重大漏洞。而这些漏洞和问题只靠征管人员去加强征管，难以实现监管和处罚。

## 1.2 国内外研究现状

信息系统和信息处理从人类文明产生开始就已存在，直到电子计算机问世、信息技术实现飞跃以及现代社会对信息需求日益增长，才迅速发展起来。从第一台电子计算机于 1946 年问世，50 多年来，信息系统经历了由单机到网络，由低级到高级，由电子数据处理到管理信息系统、再到决策支持系统。20 世纪 70 年代初随着数据技术、网络技术和科学管理方法的发展，计算机在管理上的应用日益广泛，管理信息系统逐渐成熟起来。管理信息系统最大的特点是高度集中，能将组织中的信息集中起来，进行快速处理，统一使用。有一个中心数据库和计算机网络系统是 MIS (Management Information System) 的重要标志。MIS 的处理方式是在数据库和网络基础上的分布式处理。随着计算机网络和通讯技术的发展，不仅能把组织内部的各级管理联结起来，而且能够克服地理界限，把分散在不同地区的计算机网互联，形成跨地区的各种业务信息系统和管理信息系统。

管理信息系统是 80 年代才逐渐形成的一门新学科，其概念至今尚无统一的定义。但从国外学者给 MIS 所下的定义来看，MIS 的定义也在逐渐发展和成熟。管理信息系统是信息系统在管理领域的具体应用，具有信息系统的一般属性。从管理信息系统的建立、功能等方面来分析，管理信息系统可以定义为：管理信息系统是用系统思想建立起来的，以电子计算机为基本信息处理手段，以现代通讯设备为基本传输工具，且能为管理决策提供信息服务的人机系统。即管理信息系统是一个由人和计算机等组成的，能进行管理信息的收集、传输、存储、加工、维护和使用的系统。

随着信息技术的迅猛发展，尤其是数据库技术、网络技术以及计算机硬件的快速发展，促使管理信息系统的蓬勃发展。现在已经广泛应用于工厂、企业、学校、医院、政府等社会各个领域，并朝着新的方向注定永不停歇的前进。国外经济发达国家早在六、七十年代就已将计算机应用到税收业务上。各国政府投入巨额资金，建立庞大的全国计算机网络，有统一规范的信息系统支持，各

种纳税事务的处理，都通过计算机网络进行，收到很好的效果。近年来，许多发展中国家也把税收电子化作为加强政府经济调控能力的最主要措施来抓。可以说，除税收法制健全的因素外，税务工作拥有先进的管理手段，是这些国家税收秩序良好的主要原因。在技术上，税收信息系统大都采用客户机 / 服务器 (Client/Server) 体系结构，采用先进的大型数据库和网络系统。美国从上世纪 60 年代起逐步在全国范围内建立了税收征管网络。实现了从税收预测、税务登记、纳税申报、税款征收、税务稽查、税源控制、纳税资料的收集、存储、检索等一系列工作环节的信息化。在税收信息化的建设过程中，重视先进技术的运用，是美国的一大特色。如 1999 年，美国开始运用信用卡技术，支付预估的税款；2000 年，美国开始采用顾客账户方式，纳税人通过国税局电子报税系统支付的税款可以直接从其银行账户中扣除。澳大利亚已在全国税务机关内部全面运用计算机系统管理纳税申报，办理出口退税等日常工作。并实现了与政府相关部门如海关、工商、保险、金融及大企业的网络互联，有效地对税源进行控制，有针对性地开展税务审计。另外，在安全方面，澳大利亚税务系统也采取了一些措施，如为防止灾难性毁坏而设计建立了数据库备份运行系统，以备不时之需。在保密机制上，采用了口令或密码、电子通行证等机制，同时使系统具有了屏幕保护功能、权限保护功能和追踪查询功能。在欧盟国家中，意大利拥有最成功、最大的税收信息管理系统——ITIS (Italy Tax Information System)。财政部通过 ITIS 对全国税收工作进行管理，同时，通过公用数据网实现税收环节相关部门的信息交换和资源共享。日本的税收信息管理系统也有其自身的特点。首先，国税局及税务署的系统根据征管工作的需要统一开发运行。国税局接收税务署传送的纳税人信息，并对银行传送的税款入库信息进行核对后，再传送给税务署，后者采用统一的定型统计，从而实现了国税局与税务署系统在统一的状态下运行。

我国的税收信息化的建设历程始于二十世纪八十年代。20 年来，我国的税务信息化的发展轨迹大致可划分为三个阶段<sup>[1]</sup>：第一阶段是模拟手工操作的税收电子化阶段，其总体特征为采用数据库技术，依托单机和局域网，涉及税务应用的操作层次，对税收业务的重要环节实现了手工操作的计算机化。第二阶段是步入面向管理的税收管理信息系统阶段，其总体特征为采用关系型数据库、客户机/服务器模式及图形化界面，依托广域网进行分布式处理，涉及税务应用的操作和管理层次。第三阶段是实现创造税收价值的全方位税收服务系统阶段，其总体特征为采用 WEB 技术和组件化结构，依托互联网实现集中式处理，涉及税务应用的操作、管理和决策层次，并对纳税人进行全面的管理与服务。国务院于 1994 年开始实施“金税工程”，金税一、二期的建设，建立



了覆盖省、市、县广域网并与总局构成四级网络，基础设施建设初具规模。配合信息化建设的需要，规范了税务机构及工作规程和流程。1999年，神州数码承建国家税务总局的“中国税收征管信息系统CTAIS”，目前该系统已在全国国税部门推广应用。CTAIS系统全面覆盖基层业务处理、管理监控和辅助决策等各个税收征管环节的业务。CTAIS较充分地考虑了税收应用系统的特点，功能全面，业务监控关系严密，能够满足全国统一规范的要求。计算机应用以“金税工程”为先导，逐步深入到税收征收管理、出口退税、办公自动化、财务、监察、反避税、数据分析、系统监控等各个领域。税收信息化发展已经渗透到税收工作的各个方面，规范、统一、高效的税收信息化发展，很好地实现了为国聚财、保证政府税收收入的功能，有效地维护了纳税人权益，为纳税人提供了便捷的纳税服务<sup>[6,7]</sup>。

但因为税收起着重要的经济杠杆作用，税收法律规章、税收政策的变动频繁，而有的变动具有临时杠杆作用，税收法律规章、税收政策的变动频繁，而有的变动具有临时性和特殊性，如果为此改动现有的系统将产生巨大的成本，为此又产生了新的需求，教育储蓄个人所得税管理系统就是这样产生的。而税收信息化的现状为系统的开发提供了充足的硬件、安全畅通的网络和丰富的开发经验和开发技术人员。

### 1.3 课题研究的目的和意义

由于教育储蓄规范管理工作牵涉面广，如执行不当，不但会影响征税公平，减少税收收入，也不能真正达到免征教育储蓄个人利息所得税的目的和效果，同时给各金融机构的工作带来影响，而且会造成不良的社会影响，为规范教育储蓄，国家税务总局、中国人民银行、教育部在广泛调研的基础上，联合发布了《教育储蓄存款利息所得免征个人所得税实施办法》（以后在本文简称实施办法）。

《实施办法》在《中国人民银行关于印发〈教育储蓄管理办法〉的通知》（银发〔2000〕102号）中关于教育储蓄政策规定的基础上，主要针对教育储蓄存在的虚假证明问题，规范了正在接受非义务教育的学生身份证明（以下简称“证明”）的印制、领取、开具和使用；并且，重申和明确了相关法律责任等。相对于老办法而言，并没有调整原有教育储蓄利息的税收政策，只是针对近年来一些地区利用虚假教育储蓄骗取免税优惠的问题而制定了加强管理的措施。这对于规范教育储蓄工作、进一步加强教育储蓄利息的免税管理有着积极意义。为配合实施办法的落实，为加强税收征管，推进依法治税，堵塞教育储蓄免税管理漏洞，规范教育储蓄免税管理，强化扣缴义务人履行代扣代缴义务，确保教育储蓄存款利息所得免征个人所得税政策的正确贯彻执行，必须借助计算机技术来得以实现。但现

在用的综合征管软件没有包括这一功能，现在也没有这方面的软件，在这种情况下，为了确保教育储蓄利息所得免征个人所得税规范管理工作落实到位，有效监控教育储蓄个人所得税，永州市国家税务局决定组织专门力量进行教育储蓄存款利息所得免征个人所得税管理软件的开发。

坚持科学管理和信息高度集中的原则，研究和开发出教育储蓄个人所得税管理系统。该系统作为主体征管软件的外延拓展补充，满足了教育储蓄个人所得税征收管理的实际工作需要。该系统在对纳税户的开户信息、流水帐信息、证明单信息进行调查和采集录入的基础上，通过数据比对、报表生成，实现了教育储蓄个人所得税管理的规范化，大大减少了工作量和征管成本，有效提高征管效率，实现了教育储蓄个人所得税管理的公平化、规范化和公开化。该系统不仅可以有效地解决目前教育储蓄个人所得税征收管理工作中存在的问题，实现对违规教育储蓄的有效监控，使教育储蓄个人所得税征收管理工作日趋规范、科学、合理，同时可以实现上级税务机关对下级税务机关的监控，进一步规范税务系统内部的管理，为免征教育储蓄个人所得税优惠政策的实施创造良好的外部环境。

## 1.4 论文的组织

本文的组织如下：第一章主要讨论课题的研究背景、国内外研究现状、目的和意义；第二章主要介绍了教育储蓄个人所得税管理系统的需求分析；第三章主要阐述了系统的总体设计；第四章主要对在系统开发过程中的关键技术及其实现进行了详尽的描述；第五章是对系统的测试及使用进行简单地介绍；第六章是对系统在运行过程中取得的成效进行总结并展望未来的研究方向。

## 第二章 系统需求分析

需求分析是软件开发过程的初始阶段，是一个系统的推理过程，涉及到诸多领域的问题，目的在于理解需要什么和必须做什么，并将用户的需求用书面形式表达出来，是关系到软件开发成败的关键阶段。软件产品的质量，在很大程度上取决于是否进行过系统的需求分析，并是否取得正确、一致和比较完全的需求文档。

### 2.1 教育储蓄个人所得税管理系统的总体目标

教育储蓄个人所得税管理系统的功能是将具有代扣代缴利息个人所得税义务的金融机构采集申报的教育储蓄储户信息进行分析比对，从而找出违规教育储户并进行处罚的系统。根据所提供的业务需求，结合教育储蓄管理的特点，从储户开户信息、存取款信息、证明单信息等内容的采集、比对和分析，构造一个完整的计算机信息管理系统，在比对的基础上，找出违规的储户、银行、学校，分别情况进行处理，并为管理工作提供查询统计。从而真正实现对教育储蓄管理的有效监控，满足新形势下教育储蓄政策实施的需要。采用 C/S(Client/Server)模式运行，由教育储蓄管理业务相关人员使用。系统的主要功能是通过储户开户信息、证明单信息、流水账信息等内容的采集、比对和分析，将存在的教育储蓄重复开户、超限额存取款、教育期限重复等违规情况找出来，分别各种情况进行处罚和补征利息个人所得税，实现对银行和储户的有效监控，使教育储蓄的管理日趋规范、科学、合理。同时，还提供证明单管理和查询统计功能，通过比对和各种报表的查询统计，实现上级税务机关对下级税务机关的教育储蓄个人所得税管理的监控，进一步规范税务系统内部的管理。系统具有操作便捷，使用简便，运算准确的特点。

考虑到满足教育储蓄管理的实际情况，基于 C/S 结构的教育储蓄个人所得税管理系统应满足以下的设计目标或要求：

- 1、可行性：系统必须支持并行操作，以满足国税部门不同层次或人员对系统的操作需要。
- 2、安全性：需要保证系统所有数据(包括静态和动态数据)的安全性；同时，也要保证传输数据的安全性。
- 3、远程访问：系统应该提供县一级国税局到市一级国税局甚至省一级国税局的远程访问手段。
- 4、可扩展性：系统需要有一定的扩展性，以满足不断更新的政策及形势需要。

5、可靠性和可用性：系统需要在一定的负荷下保持运行的可靠性和可用性。

6、接口服务：系统应具有与银行的接口服务功能，接收银行申报数据导入系统。

## 2.2 教育储蓄个人所得税管理业务要求

根据《实施办法》和相关法律法规，教育储蓄管理相关的主要业务要求有<sup>[2]</sup>：

1、按照有关规定，开立教育储蓄的对象必须是中国大陆在校小学 4 年级（含 4 年级）以上学生；享受免征利息税优惠政策的对象必须是正在接受非义务教育的在校学生，其在就读全日制高中（中专）、大专和大学本科、硕士和博士研究生三个阶段中，每个学习阶段可分别享受一次 2 万元教育储蓄的免税和利率优惠。也就是说，一个人至多可以享受三次优惠。

2、办理储蓄存款业务的各商业银行和城乡信用社（不含邮政储蓄机构）均可开办教育储蓄。

3、教育储蓄采用实名制。办理开户时，须凭符合条件的学生本人户口簿（户籍证明）或居民身份证等有效实名证件到储蓄机构以学生本人的姓名开立存款账户。金融机构根据储户（即学生）提供的上述证明，登记证件名称及号码等事项。代理学生办理时，代理人还必须同时出具自己的有效身份证明。

4、教育储蓄为一年、三年和六年期零存整取定期储蓄存款。最低起存金额为 50 元，每月固定存额（存额由储户自定），分月存入，中途如有漏存，应在次月补齐，未补存者到期支取时按实际存额和实际存期计算利息。每份本金合计不得超过 2 万元。每份本金合计超过 2 万元或一次性趸存 2 万元的，一律不得享受教育储蓄免税的优惠政策，其取得的利息，应征收利息税。不按规定计付利息的教育储蓄，不得享受免税优惠，应按支付的利息全额征收利息税。同时，教育储蓄实行利率优惠。一年期、三年期教育储蓄按开户日同期同档次整存整取定期储蓄存款利率计息；六年期按开户日五年期整存整取定期储蓄存款利率计息。教育储蓄到期时，按实存金额和实际存期计算利息。教育储蓄在存期内遇利率调整，仍按开户日利率计息。

5、教育储蓄到期时，储户必须持存折、身份证或户口簿（户籍证明）和学校提供的“证明”一次支取本金和利息。储户凭“证明”可以享受利率优惠，并免征利息税。储蓄机构应认真审核储户所持存折、身份证或户口簿（户籍证明）和“证明”，对符合条件的，给予免税优惠，并在“证明”（第二、三联）上加盖“已享受教育储蓄优惠”印章，每份“证明”只享受一次优惠。代理人代为支取时，还必须同时持代理人的身份证等有效实名证件及复印件（复印件留储蓄机构）。

6、对不能提供“证明”的储户，其教育储蓄不享受利率优惠。即一年期、三年期按开户日同期同档次零存整取定期储蓄存款利率计付利息；六年期按开户日五年期零存整取定期储蓄存款利率计付利息。同时，其利息所得应按有关规定扣缴利息税。

7、教育储蓄提前支取时必须全额支取。提前支取时，储户能提供“证明”的，仍可享受利率和免税的优惠。即按开户日挂牌公告的与实际存期相对应的同期同档次整存整取定期储蓄存款利率计付利息，并免征利息税。对储户未能提供“证明”的，按实际存期和支取日活期储蓄存款利率计付利息，并按有关规定征收利息税。

逾期支取教育储蓄，其超过原定存期的部分，按支取日活期储蓄存款利率计付利息，并对逾期期间的利息征收利息税。

8、教育储蓄到期前，储户必须持存折（或复印件，复印件仅适用于在外地读书的学生）、户口簿（户籍证明）或身份证到所在学校开具“证明”。“证明”样式由国家税务总局制定，各省、自治区、直辖市和计划单列市国家税务局印制，由学校到所在地主管税务机关领取。“证明”一式三联，第一联学校留存；第二、三联由储户在支取本息时提供给储蓄机构；储蓄机构应将第二联留存备查，第三联在每月办理扣缴税申报时一并报送主管税务机关。

从事非义务教育的学校应主动向所在地国税机关领取“证明”，并严格按照规定填开“证明”，不得重复填开或虚开，对填开的“证明”必须建立备案存查制度。储户到所在学校开具“证明”时，应在“证明”中填列本人居民身份证号码；无居民身份证号码的，应持本人户口簿（户籍证明）复印件三份，分别附在三联“证明”之后。

9、储蓄机构应对教育储蓄情况单设会计科目反映，进行详细记录以备税务机关核查，并按月统计逐级上报。记录的内容应包括：储户姓名、证件名称及号码、开具“证明”的学校、“证明”编号、存款额度、储蓄起止日期、利率、利息。

10、主管税务机关应设立教育储蓄利息所得免征个人所得税台账，对储户享受优惠情况进行详细登记。登记内容包括：储户姓名、证件名称及号码、“证明”编号、开具“证明”的学校、开户银行、存款额度、储蓄起止日期、利率、利息。

主管税务机关应依法定期对储蓄机构的教育储蓄存款利息所得免税情况开展检查。

- 11、学生到异地（仅指中国大陆境内）上学，只要凭接受非义务教育学校所在地省级国家税务局统一印制、所在学校开具的“证明”，到开立教育储蓄账户的银行按规定办理即可。
- 12、对违反规定向纳税人、扣缴义务人提供“证明”，导致未缴、少缴个人所得税款的学校，按《中华人民共和国税收征收管理法》（以下简称《征管法》）及其实施细则的规定，税务机关对其处未缴、少缴税款 1 倍以下的罚款。对储蓄机构以教育储蓄名义进行揽储，没有按规定办理教育储蓄，而造成应扣未扣税款的，按《征管法》的规定，向纳税人追缴应纳税款，并对扣缴义务人处应扣未扣税款 50% 以上 3 倍以下的罚款。税务机关在向纳税人追缴税款时,可责成扣缴义务人从纳税人的储蓄账户上限期补扣应扣未扣的税款。
- 对储户采取欺骗手段办理教育储蓄的，一经发现，对其征收利息税，并按《征管法》的规定予以处理。

2.3 系统的基本业务流程

教育储蓄个人所得税管理系统的基本业务流程是：数据采集→导出导入→数据比对→违规处理→监控管理和查询统计。其系统流程图如图 2-1 所示：

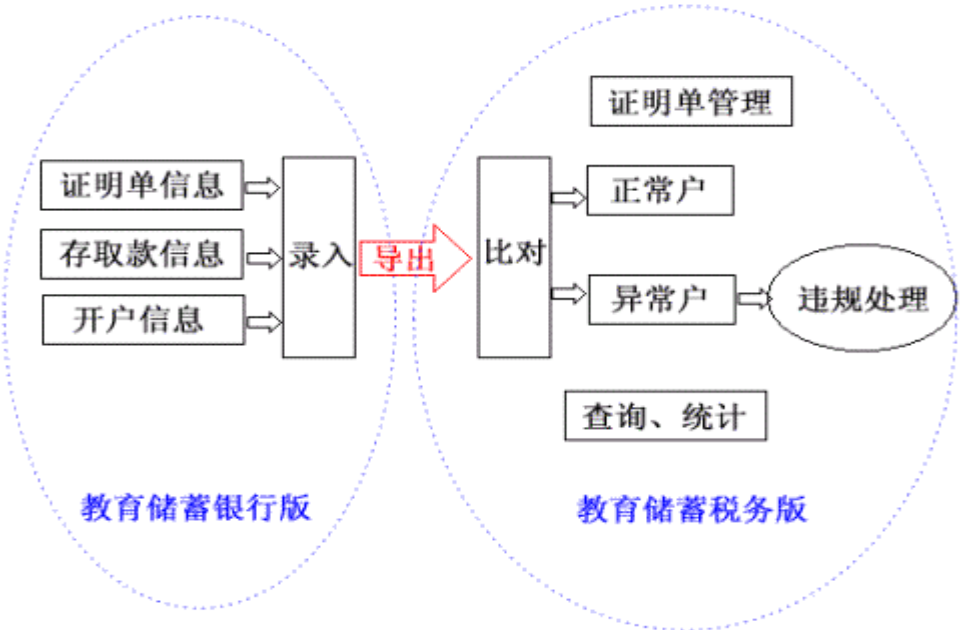


图 2-1 系统流程图

1、数据采集是推广应用教育储蓄个人所得税管理系统的基础。各软件应用单位应加强与各金融机构的沟通与管理，确保银行开户信息、流水账信息、证明单信息的采集质量、完整性和及时性。并且先要对所有全市所有金融机构进行一次

摸底，进行编码。

2、导出导入是数据采集和数据比对的桥梁。各金融机构采集相关信息后导出产生文本文件，然后在系统中进行导入构成数据比对的基础。

3、数据比对是教育储蓄个人所得税管理系统运行的核心。按首银行申报数据中的储户开户、存取款、证明单信息，将申报的数据进行归并，然后再以银行代码+银行帐号、储户开户登记的身份证/户口本号+储户姓名为关键字进行相关大比对查询，将查询得出的重复开户、超限额存取款、教育期限重复等违规信息写入违规登记表，在日常处理中进行处理。

4、监控管理和查询统计主要是根据教育储蓄征收管理工作的实际需要，对教育储蓄管理的基本工作情况进行监控管理和查询统计。监控管理主要完成对证明单及违规情况等方面的管理。查询统计主要完成各种工作中需要的报表的查询。

2.4 系统数据流程图

对于系统数据的流向首先是由银行经办人员采集数据然后导出报送税务机关征管部门，税务机关系统操作员将数据导入系统，这是数据在系统中的入口，然后形成储户开户信息表、存取款信息表、证明单信息表，通过数据比对程序产生异常信息表；最后根据分析确定各种违规情况进行相应处理并产生处理情况表。系统顶层数据流程图如图 2-2 所示，系统详细数据流程图如图 2-3 所示。

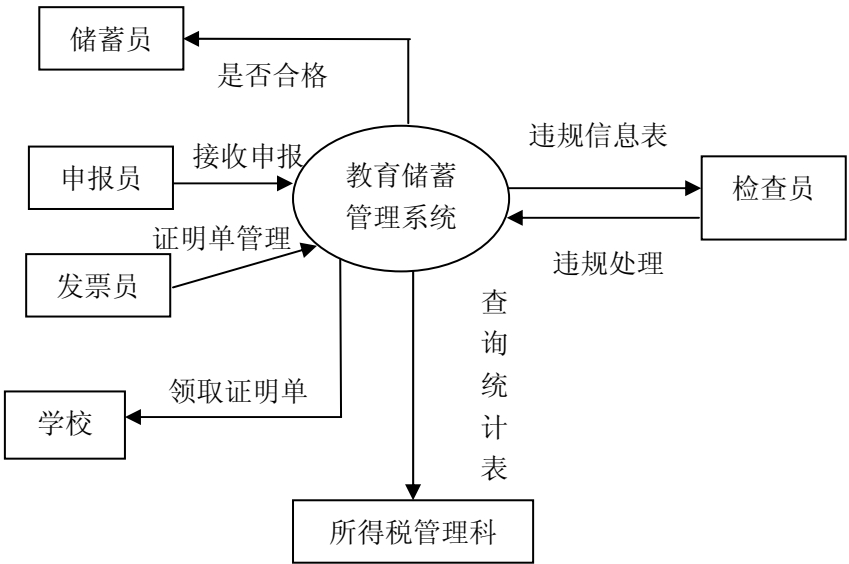


图 2-2 顶层数据流程图

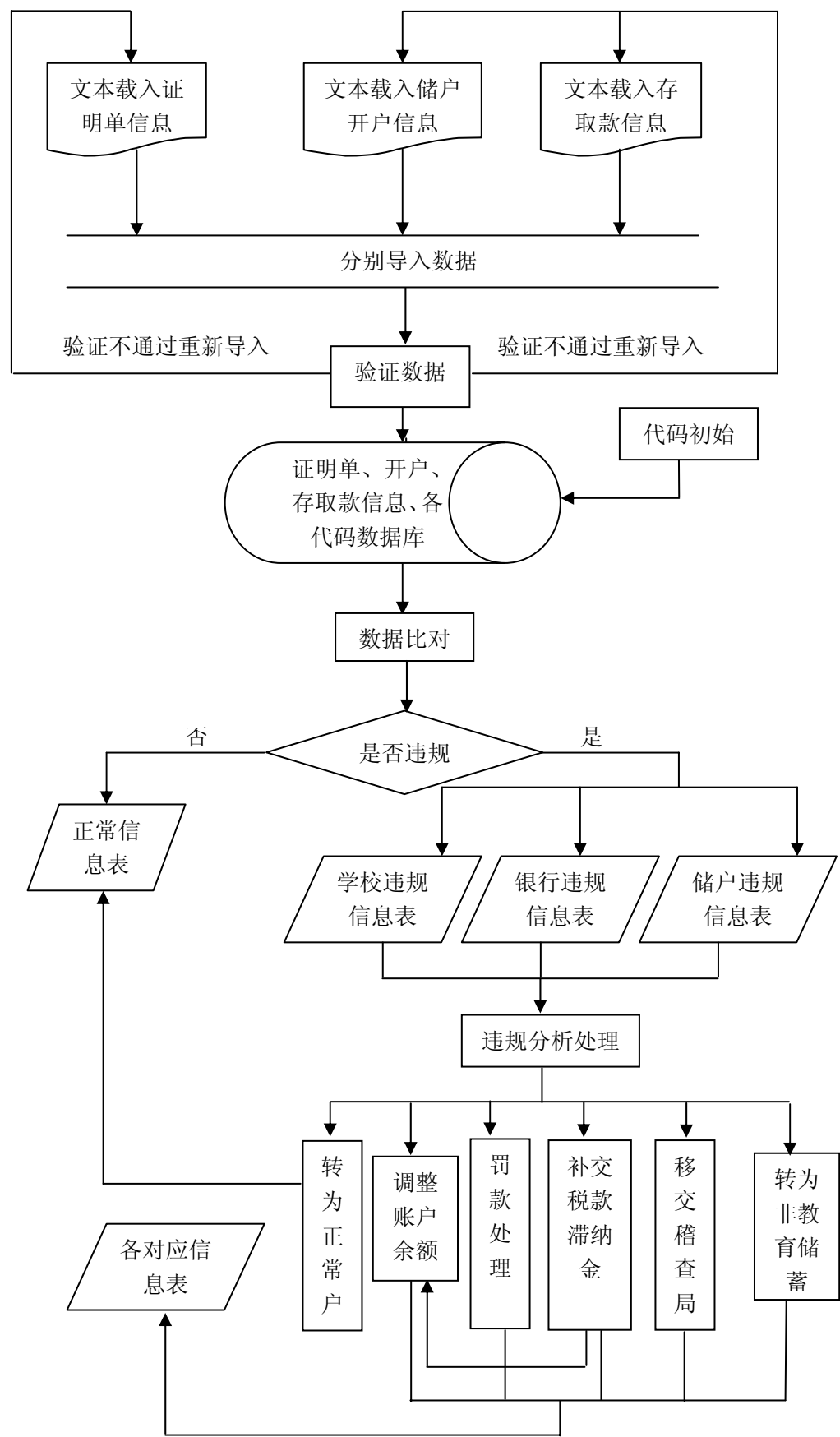


图 2-3 系统数据流程图



## 2.5 本章小结

本章简单的说明了系统的需求分析，包括系统的总体目标、业务要求，并以图例的方式说明了系统的业务流程和数据流程。

## 第三章 系统总体设计

设计和开发一个管理信息系统，首先进行系统需求分析，然后是进行系统总体设计。这一章的主要目标是对系统运行的网络结构、硬件环境、软件环境、系统结构图、系统模块结构及实现的主要功能等方面的内容做出说明。

### 3.1 系统的网络结构

全国国税系统税收信息化建设迅速发展，金税一、二期的建设，建立了覆盖省、市、县广域网并与总局构成四级网络。2005 年 1 月 1 日综合征管软件——CTAIS（中国税收管理信息系统）在全省推广应用，对全省的广域网进行了全面 SDH 网络改造，我们所开发的教育储蓄个人所得税管理系统，正是主要依托于已构建的这个网络体系，其网络结构示意图如图 3-1 所示：

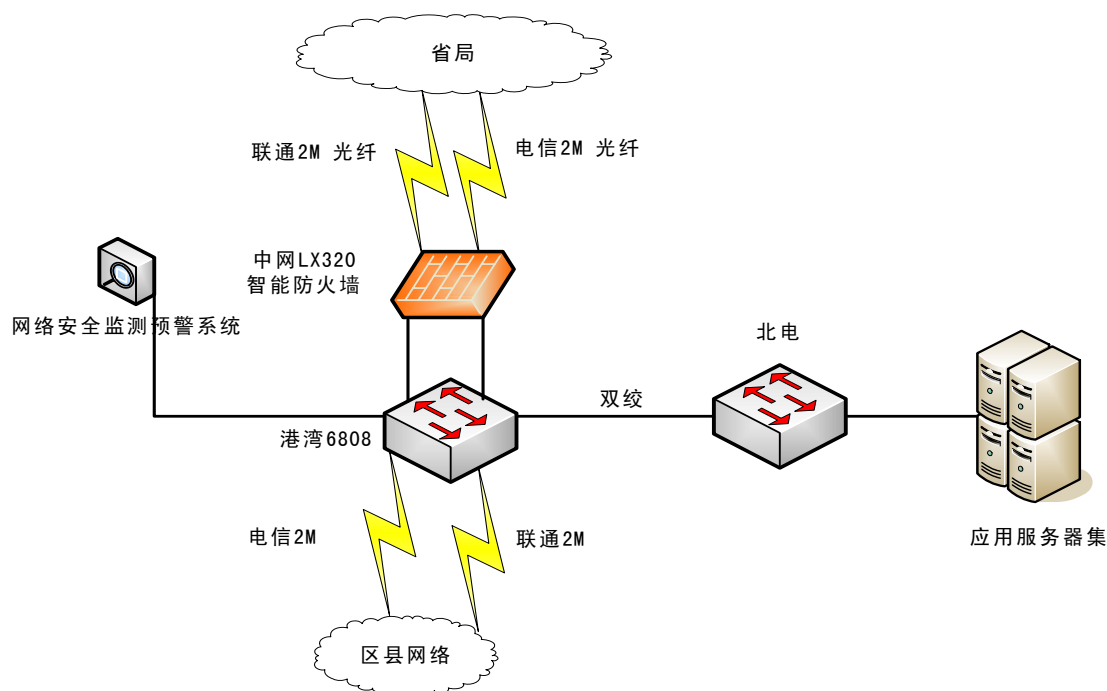


图 3-1 网络结构示意图

该系统在永州市内部网中运行，数据库服务器中存放与本系统相关数据，安装于市局，防火墙连接到市局交换机上，通过港湾 6808 路由器连接到电信 SDH 网络及备份线路联通 SDH 网络上，再通过港湾 5210 路由器连接到各县区局交换机上，客户端连接到交换机上。

## 3.2 系统运行环境

系统采用 C/S 的软件结构, 需要服务器来运行后台服务, 进行数据逻辑处理、接受并应答客户机的请求, 客户端执行前台功能, 进行显示逻辑和事务逻辑处理并提出请求。系统需要运行在一定软硬件及网络环境下, 主要是采用 PowerBuilder 9.0+SQL Server 2000 的二层结构, 使用 TCP/IP 协议。

### 1、软件环境

服务器: MS Windows 2000 Server(中文版)或 MS Windows NT 4.0(中文版);

客户端: Windows 95/98/XP/2000 或 Windows NT;

数据库服务器: Microsoft SQL Server 2000;

开发平台: PowerBuilder 9.0;

客户端通过 SQLOLEDB 方式访问数据库。

### 2、硬件环境

服务器: 处理器 Intel®Pentium®III Xeon™2.0GHz; 内存 1G; 硬盘 40G\*5;  
UPS 电源; 双网卡 10/100Mbps 自适应快速以太网卡。

客户机: 处理器 Intel®Pentium®III 800MHz; 内存 128M; 硬盘 20G;  
10/100Mbps 自适应快速以太网卡以上配置 PC 机多台;

设备拓扑结构: 系统的设备拓扑结构如图 3-2 所示。

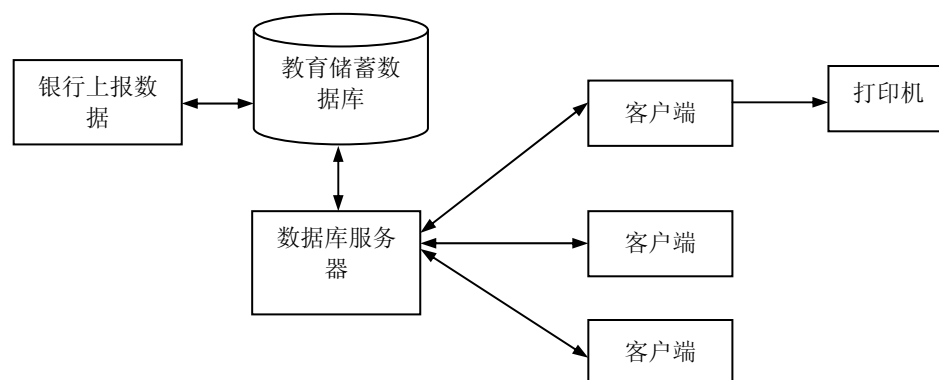


图 3-2 设备拓扑结构

## 3.3 软件功能描述

根据系统的需求分析和业务要求教育储蓄个人所得税管理系统要实现的功能主要有: 证明单管理、数据比对、数据采集、查询统计、系统设置。为此, 我们设计了如图 3-3 所示的主要程序模块。

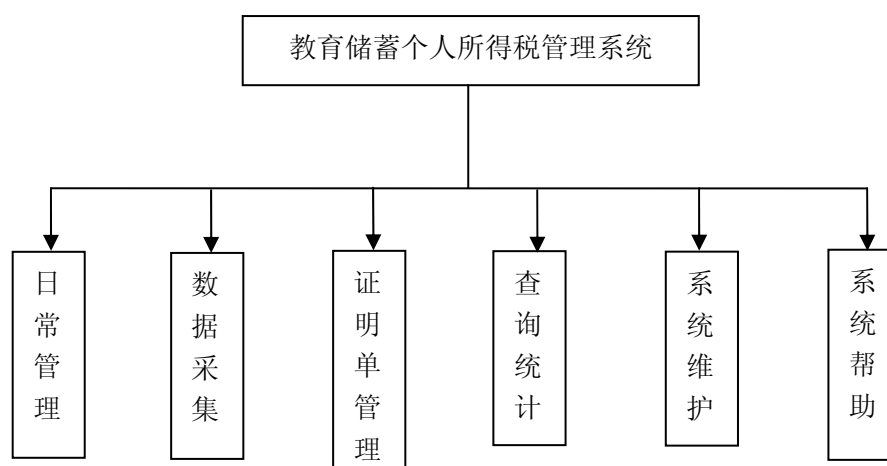


图 3-3 系统程序模块结构图

### 3.4 系统模块结构

软件系统功能设计，根据教育储蓄管理相关政策和要求，通过对教育储蓄管理的特点分析，进行系统细分，为征收管理提供科学依据；提高工作效率和工作质量；通过分析比对查出违规，避免利息个人所得税的流失，对主体征管软件是一个有益的补充，从而进一步提高征管质量。

教育储蓄个人所得税管理系统功能模块如图 3-4 所示。

教育储蓄个人所得税管理系统运行于各客户端，主要完成对教育储蓄信息的导入采集、分析比对、处理等，共分四大功能模块：日常管理、数据采集、查询统计、系统维护。

#### 1、日常管理

包括证明单管理、违规处理和数据比对。证明单的入库、发放及退回，市局票证管理员把从省局领来的证明单进行证明单入库，然后发放给各县区局票证管理员，县区局票证员将证明单入库，发放给各实施义务教育的学校，空白的未使用证明单可以退回。对采集的数据进行比对分析找出违规的储户、银行和学校进行相应的违规处理。根据具体违规类别分别作出不同的处理, 具体如表 3-1 所示：

#### 2、数据采集

将教育储蓄个人所得税管理系统银行申报采集导出的证明单信息、开户信息、存取款流水账信息进行导入，生成相关数据库表和采集情况表；导入时进行数据完整性和准确性的验证；还可以进行已申报和未申报，已采集情况的查询。

#### 3、查询统计

包括教育储蓄余额明细表、教育储蓄余额汇总表、银行违规情况明细表、学

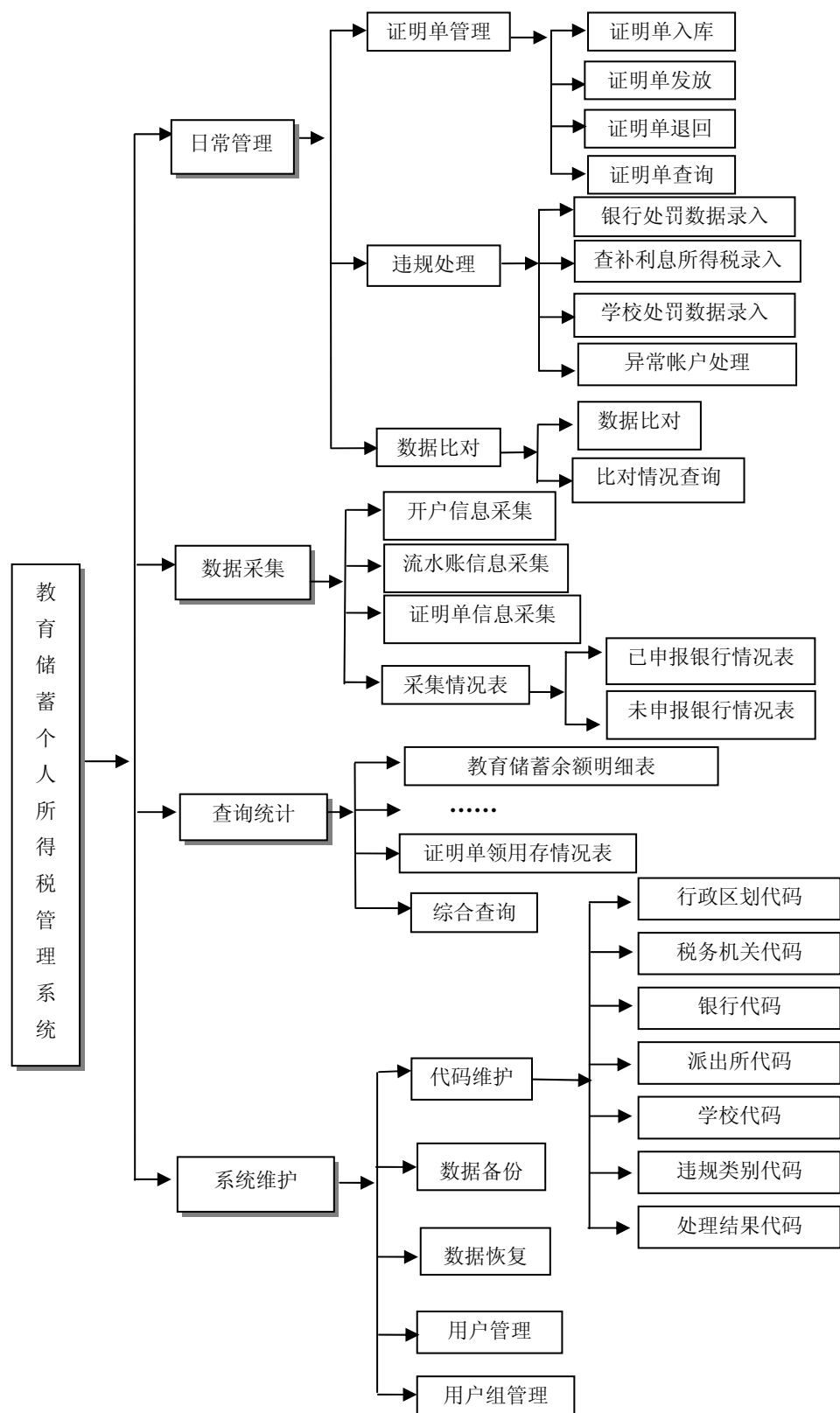


图 3-4 系统功能模块图

表 3-1 违规处理表

违规代码	违规描述	处理代码	处理描述	步数描述	所属用户组	下一用户组
01	非教育储蓄	01	银行录入错误	转为正常户	04	
01	非教育储蓄	02	银行已将该帐户设为非教育储蓄帐户	转为非教育储蓄帐户	04	
03	同行重开户	01	银行录入错误	转为正常户	04	
03	同行重开户	02	存款额最大帐户转为正常户	转为正常户	04	
03	同行重开户	03	银行已将该帐户设为非教育储蓄帐户	转为非教育储蓄帐户	04	
04	跨行重开户	01	银行录入错误	转为正常户	04	
04	跨行重开户	02	最先存款帐户转为正常户	转为正常户	04	
04	跨行重开户	03	银行已将该帐户设为非教育储蓄帐户	转为非教育储蓄帐户	04	
05	单次存超 2 万	01	银行录入错误	转为正常户	04	
05	单次存超 2 万	02	银行已将该帐户设为非教育储蓄帐户，并补交利息所得税	转为非教育储蓄帐户	04	
06	累计存超 2 万	01	银行录入错误	转为正常户	04	
06	累计存超 2 万	02	只能享受两万元的免税政策，调整帐户余额为两万元	转为正常户、调整帐户余额	04	
08	累计取超 2 万	01	移交稽查局处理	移交稽查局处理	04	09
08	累计取超 2 万	01	银行录入错误	转为正常户	09	
08	累计取超 2 万	02	查补税款	转为非教育储蓄帐户	09	
09	取款无证明单	01	移交稽查局处理	移交稽查局处理	04	09
09	取款无证明单	01	银行录入错误	转为正常户	09	
09	取款无证明单	02	查补税款	转为非教育储蓄帐户	09	
10	证明单无效	01	移交稽查局处理	移交稽查局处理	04	09
10	证明单无效	01	银行录入错误	转为正常户	09	
10	证明单无效	02	查补税款	转为非教育储蓄帐户	09	
11	证明单重复	01	移交稽查局处理	移交稽查局处理	04	09
11	证明单重复	01	银行录入错误	转为正常户	09	
11	证明单重复	02	查补税款	转为非教育储蓄帐户	09	
12	教育期重复	01	移交稽查局处理	移交稽查局处理	04	09
12	教育期重复	01	银行录入错误	转为正常户	09	
12	教育期重复	02	查补税款	转为非教育储蓄帐户	09	

校违规情况明细表、违规情况通用查询、税务机关处理情况表、利息所得税查补情况表、银行违规处罚情况表、学校违规处罚情况表等报表以及根据需求进行综合查询而产生的查询统计。

#### 4、系统维护

系统初始化的代码维护，用户和用户组权限的维护。使用教育储蓄个人所得税管理系统之前，首先要启动系统维护功能，设置好系统的初始化数据信息，包括行政区划代码、税务机关代码、银行代码、派出所代码、学校代码、违规类别代码、处理结果代码、用户组设置、用户管理等初始信息的设置及维护。还包括数据的备份和恢复功能，账号锁定和解锁等功能。

### 3.5 本章小结

本章对系统的总体设计进行了说明，分别描述了系统运行的网络结构、运行环境，对系统的功能模块及其实现的功能进行了简单的说明，最后用功能模块图展现了系统实现的主要功能模块。

## 第四章 系统关键技术及其实现

一个系统的优劣程度，可以从系统的反应速度、执行效率、安全性和易维护等方面来综合评价。而采用合适的开发模型及开发环境，规范化地数据库设计，采用并合理应用信息编码技术、数据库访问技术、系统性能优化以及安全技术等，无疑是达到系统设计要求的有効手段。本章通过对教育储蓄个人所得税管理系统开发模型和系统采用的技术方案、数据库设计等关键技术进行详细的说明，并对信息编码技术、数据库数据访问技术、存储过程、系统性能优化以及数据安全技术等系统中的具体实现进行详细的阐述。

### 4.1 开发模型

开发模型又称软件体系结构，软件体系结构的概念最初由 E.W.Dijkstra 在 1968 年提出的，当时他用分层的体系结构设计了 THE 操作系统，并提出人们应该更加关注软件是如何分解与组合的，而不仅限于编程，这样更有利于软件的开发与维护。目前软件体系结构已发展成为软件工程领域的一个独立的分支，是软件工程的一个新的研究方向。体系结构设计是大中型应用系统开发不可或缺的一环。应用框架及其基础构件独立于行业应用，能够独自演化。不同行业的应用系统可以共享统一成熟的框架系统；选择既有的框架系统，是实现软件工厂流水线生产模式的前提。应用框架的构建能力是制定行业技术标准的基础<sup>[3,4]</sup>。

现在最流行的软件开发模型是 B/S 体系结构和 C/S 体系结构，B/S 体系结构如日中天，C/S 在企业应用中依然占据着很大的市场。下面，我们分析 C/S 模式、B/S 模式的优势与不足，以利于提出适合教育储蓄个人所得税管理系统的开发模型<sup>[19,20,21]</sup>。

#### （1）C/S 模式的实用与缺陷

C/S 模式是 90 年代才发展起来的较为先进的结构，具有强大的数据操纵和事务处理能力，以及数据的安全性和完整性约束。但随着企业规模的日益扩大，应用程序的复杂程度不断提高，逐渐暴露出以下缺点：

- ◆ 缺乏系统性和具有前瞻性的结构框架；
- ◆ 封闭式单项单系统，不同系统无法交流；
- ◆ 开发成本较高。C/S 结构对客户端软、硬件要求较高，尤其是软件的不断升级，对硬件要求不断提高，增加了整个系统的成本，客户端越来越臃肿；
- ◆ 移植困难，同时维护复杂，升级比较麻烦；
- ◆ 新技术不能轻易应用。因为一个软件平台及开发工具一旦选定，不可能轻易更改，扩展性受到限制。



## (2) B/S 模式的先进与不足

B/S 模式是基于 C/S 模式发展起来的,具有操作简便、开放性好、易于维护和升级的优点。另外, B/S 模式特别适用于网上信息发布,使得传统的 MIS 功能有所发展<sup>[20]</sup>。但是也存在着不足之处:

- ◆ 由于其很好的开放性,因此,表现出安全难以控制。
- ◆ 由于其数据的集中式存放,因此,表现出网络开销很大。
- ◆ 如果网络的传输量大的话,将对带宽有一定的要求。

C/S 模式与 B/S 模式各有优势与不足: C/S 模式可实现分布的处理与集中的数据管理相结合,数据安全性强,通信速度快,但在远程数据管理方面存在不足; B/S 模式则有利于实现系统集成与扩展,可提供远程数据服务和管理,但在数据的交互性、动态服务、统计分析和图表显示方面尚有欠缺。因此,在构建信息系统模式时应根据具体的应用而决定。在安全性要求高、交互性强、处理数据量大、查询灵活、地点固定的小范围内使用 C/S 模式,在安全性和交互性要求不高,地点灵活的广域范围内使用 B/S 模式。有时根据需要还可结合两种模式的长处组合成混合模式的结构体系,开发出安全可靠、灵活方便、效率高的软件系统。

经过几十年的软件开发实践,今天所开发的系统或多或少地借鉴了以往开发的经验,解决任何新的问题并不一定需要从头做起,我们更容易复用以往成功的经验,更好地被人理解,更好、更快地完成系统的设计,开发。以往的税务应用系统大多采用的是 Client/Server 结构,是非常流行的一个风格。而教育储蓄个人所得税管理系统因其在安全性要求高、交互性强、地点固定等特点, C/S 模式更适合它的开发,因此选择了 C/S 模式作为开发模型。

C/S 是指 Client/Server(客户机/服务器), C/S 结构是近几年非常受欢迎的一种分布式计算模式。C/S 结构主要是指将一个数据库应用系统分解为前台的客户(前端)应用程序和后台的服务器部分,通过网络连接应用程序与服务器,实现数据交换。

它的优势在于广泛地采用了网络技术<sup>[12]</sup>,将系统中的各部分任务分配给分布在网络上的担任不同角色计算机,它把较复杂的计算和管理任务交给网络上的高档机器—服务器(Server),而把一些频繁与用户打交道的任务交给前端较简单的计算机—客户机(Client),通过这种结构完全实现了网络上信息资源的共享。

在 C/S 系统中,将应用程序分为两大部分:一部分是由多个用户共享的信息与功能,这部分称为服务器;另一部分是为每个用户所专有,称为客户部分。客户部分负责执行前台功能,如管理用户接口、数据处理和报告请求等;而服务器部分执行后台服务,如管理共享外设、控制对共享数据库的操纵、接受并应答客户机的请求等。

这种结构的核心是客户应用程序仅发送服务请求,一切由服务器完成,结果

发回客户应用程序，而服务器所进行的工作对客户应用程序是完全透明的。这种体系结构由多台计算机分别执行，使它们有机地结合在一起，协同完成整个系统的应用，从而达到系统中软、硬件资源最大限度的利用。

C/S 应用系统基本运行关系体现为“请求/响应”的应答模式。每当用户需要访问服务器时就由客户机发出“请求”，服务器接受“请求”并“响应”，然后执行相应的服务，把执行结果送回客户机，由它进一步处理后再提交给用户。

任何一个应用系统，从简单的单机系统到复杂的网络计算<sup>[4]</sup>，它都由三部分组成：显示逻辑部分(表示层)，事务处理逻辑部分(功能层)和数据处理逻辑部分(数据层)。表示层的功能是实现与用户的交互；功能层的功能是进行具体的运算和数据的处理；数据层的功能是实现对数据库中的数据进行查询、修改、更新等任务。由于 C/S 结构被设计成两层模式，显示逻辑和事务处理逻辑均被放在客户端，数据处理逻辑和数据库放在服务器端，从而使客户端变的很“胖”，成为胖客户机，服务器端的任务相对较轻，成为瘦服务器。C/S 数据库系统结构如图 4-1 所示。

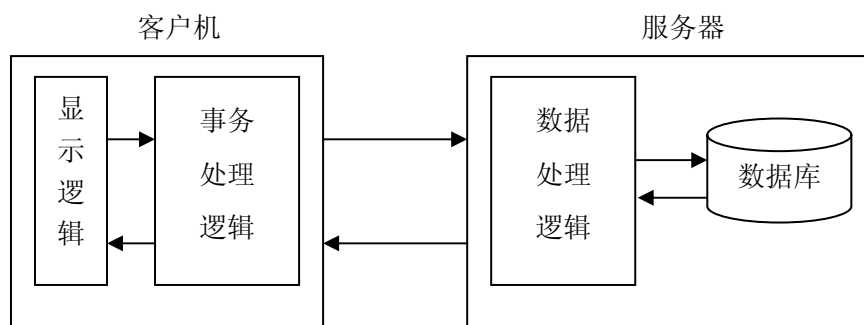


图 4-1 C/S 数据库系统结构

基于 C/S 模式下的教育储蓄个人所得税管理系统，综合考虑系统的总体目标、软硬件环境与系统开发、维护周期及工作效率等因素，采用的方案如下：操作系统采用基于 NT 技术开发的 Windows 2000 Server，后台数据库采用 SQL Server 2000，前台开发工具采用 Powerbuilder9.0。

Microsoft Windows2000 Server 这个版本以前的名称是 Windows NT SERVER 5.0，是为服务器开发的多用途操作系统，是一个性能更好、工作更稳定、更容易管理的平台<sup>[15]</sup>。

该系统后台数据库采用微软公司出品的 SQL Server 2000，它是目前常用的网络数据库之一<sup>[37]</sup>，它性能可靠、功能强大。SQL Server 2000 是一个真正的基于客户/服务体系结构的 RDBMS 产品。SQL Server 服务器软件主要由两大部份组成，即 SQL Server 2000 内核(Kernel)及数据库管理系统(DBMS)。其中 DBMS 部份完成数据存取，资源管理、安全控制等各项操作。SQL 内核部分处理客户与 SQL Server 2000 的连接及磁盘和网络 I/O 等，它负责 DBMS 提供必要的运行环境。在它所提

供的众多数据处理工具中，存储过程技术以其强大的功能、卓越的灵活性而成为佼佼者，备受数据库开发人员的青睐。在数据库应用系统中恰当而充分地使用存储过程不仅能向用户返回数据、在表中插入修改数据、执行系统函数和管理任务，而且能自动地在 SQL Server 2000 上存储应用程序的业务逻辑和函数功能，这样在服务器级上就可以对数据做出有关决策，而不是把结果返回给客户机。总之，它们是 SQL Server 2000 中功能非常强大的一部分，它能使开发工作更轻松<sup>[38,39]</sup>。

该系统前端开发工具是 Powerbuilder 9.0。自 1991 年由 POWERSOFT 公司推出数据库应用开发工具 Powerbuilder 以来，它已逐渐成为开发客户机/服务器体系应用程序的首选开发环境。它是一种面向对象的、具有可视图形界面的、快速的交互式环境。在数据库应用程序开发方面，Powerbuilder 一直处于领先地位。随着数据库技术在各行各业为广泛地应用，作为企业级数据库前端开发工具 Powerbuilder 越来越广泛地应用于开发与数据库相关的大中型应用系统<sup>[25]</sup>。

4.2 数据库设计

4.2.1 设计原则

设计一个数据库需要经历一个从现实环境到抽象模型，从抽象模型到具体实现的过程。要完成这一过程，就必须依靠各种不同的数据模型。数据模型就是对现实世界进行抽象的工具。在信息管理中需要将现实世界的事物及其有关特征转换为信息世界的的数据，才能对信息进行处理和管理，这就需要依靠数据模型作为这种转换的桥梁。这种转换经历了从现实到概念模型，从概念模型到逻辑模型，从逻辑模型到物理模型的转换。这几种数据模型的关系参见图 4-2。

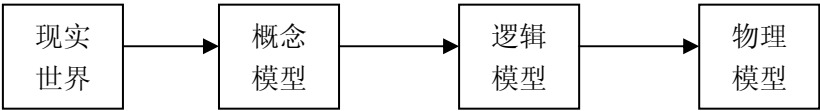


图 4-2 数据库设计的模型

数据库设计是应用程序设计的基础，其性能直接影响应用程序的性能。数据库性能包括存储空间需求量的大小和查询响应时间的长短两个方面。为了优化数据库性能，需要对数据库中的表进行规范化。规范化就是满足适当范式的模式，通常可以通过判断分解后的模式达到几范式来评价模式规范化。范式级别从低到

高有：1NF、2NF、3NF、BCNF、4NF、5 范式，通过分解，可以将一个低一级范式的关系模式转换成若干个高一级范式的关系模式。产生冗余和异常的两个主要原因是部分依赖和传递依赖。因为 3NF 模式中不存在非主属性对码的部分函数依赖和传递函数依赖，所以具有较好的性能。数据库模式通常要变换成为 3NF 或更高级别的范式。1972 年，Dr.E.F.Codd 在他的论文“进一步正规化数据库的关系模型中”就提出了数据库从第一范式到第三范式的数据正规化设计技巧。第一范式，没有重复的组或多值的列；第二范式，每个非关键字段必须依赖于主关键字，不能依赖于一个组合式主关键字的某些部分；第三范式，一个非关键字段不能依赖于另一个非关键字段。虽然一个比一个的规范化程度更高，但是满足第三范式的表结构容易维护且基本满足实际应用的要求。因此，一般都按照第三范式的标准进行规范化。

规范化的优点是：减低了数据冗余度，从而减少数据库的存储空间；保证数据增加、删除和修改等到操作的有效正常进行，维护了数据的一致性。但是规范化也有缺点，即由于将一个表拆分成为多个表，在查询时需要多表连接，从而降低了查询速度。基于这个缺点，加上存储成本的日益降低，同时考虑到一些应用需要较快的响应速度，综合考虑权衡存储与性能，在设计表时应考虑对某些表采取分割表、保留冗余列、增加派生列等方法进行反规范化。

数据库设计的另一个重要问题就是完整性约束，完整性规则提供了一种手段来保证当授权用户对数据库作修改时不会破坏数据的一致性。因此，完整性规则防止的是对数据的意外破坏。关系模型的完整性规则是对关系的某种约束条件。关系的完整性共分为三类<sup>[39]</sup>：

（1）实体完整性（entity integrity）：规定基本关系 R 的主属性不能取空值。在实际数据库设计中是通过定义主码又称主键（primary key）来实现。

（2）参照的完整性（referencial integrity）：现实世界中的实体之间往往存在某种联系，在关系模型中实体及实体间的联系是用关系来描述的，这样自然就存在着关系与关系间的引用。在实际数据库设计中是通过定义外码（foriegn key）来实现的。

（3）用户定义的完整性（user define integrity）：就是针对某一具体的关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求，由应用

的环境决定。如教育储蓄个人所得税管理系统中银行的用户账号必须为八位。

在教育储蓄个人所得税管理系统中对相关数据表建立了主键、索引和外键，以及 unique、not null、默认值等属性使整个数据库的实体完整性、引用完整性和用户自定义完整性得到了有效的保证。

4.2.2 数据建模

在需求分析阶段与用户的反复交流，得到收集的数据元素列表：储户名，证件类型，证件号码，开户银行，储蓄类别，存款金额，取款金额，利息金额，证明单编号，学校名，行政区划，所属税务机关.....这些数据元素属于储户、银行、学校、证明单、操作员等实体。

创建了所属实体后，就可以确定实体间的联系。教育储蓄个人所得税管理系统的实体联系（ER）图见图 4-3：

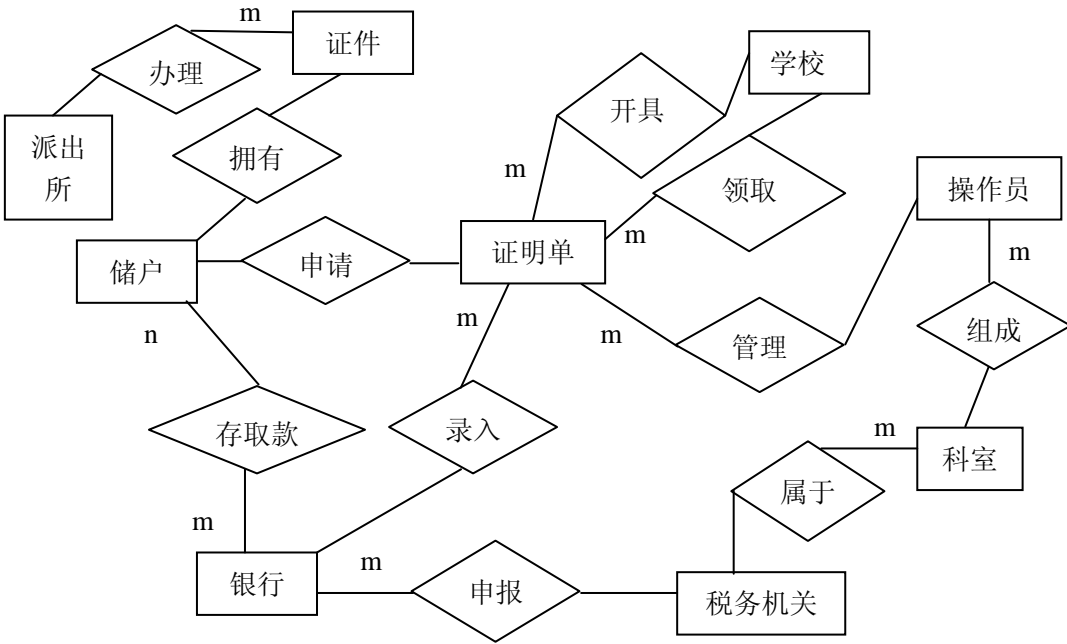


图 4-3 ER 图

根据转换规则，该 ER 图中的实体将转换成关系模式，联系也将转换成关系模式集。经过权衡存储空间与性能以及应用的需要，进行规范化与反规范化后。具体成以下主要实体类型及联系类型，其中主键用\*号标识，外键用下划线标识：

银行（\*银行代码，银行名称，上级银行代码，所属税务机关代码，银行地址，联系电话，申报人姓名）

学校（\*学校代码，学校名称，学校类别代码，学校地址，所属税务机关代码，

开票人，联系电话)

学校类别 (\*学校类别代码, 学校类别)

储户 (\*证件号码, 证件类别代码, 储户姓名, 所属派出所代码, 银行代码, 银行账号, 储蓄类别代码, 存期, 存款金额, 取款金额, 证明单编号, 利息金额, 开户日期, 到期日期, 销户日期)

操作员 (\*用户代码, 用户名, 用户口令, 所属税务机关代码, 用户组代码, 证明单出入库管理标志, 违规处理标志, 数据比对标志, 代码维护标志, 用户管理标志, 用户组管理标志, 数据查询标志, 数据统计标志)

证明单 (\*证明单编号, 学校代码, 证件类别代码, 证件号码, 派出所代码, 储户姓名, 银行代码, 银行名称, 银行账号, 学生姓名, 学校类别, 开具人, 开具日期)

证件 (\*证件类别代码, 证件类别)

税务机关 (\*本级税务机关代码, 本级税务机关名称, 上级税务机关代码, 上级税务机关名称, 税务机关级别, 行政区代码)

储蓄类别 (\*储蓄类别代码, 储蓄类别描述)

派出所 (\*派出所代码, 派出所名称, 派出所地址, 行政区代码)

行政区 (\*行政区代码, 行政区名称)

存取款 (银行代码, \*银行帐号, 操作日期, 存取标志, 存取金额, 储蓄类别代码, 利息金额, 所属时间, 操作员)

开户 (\*银行帐号, 银行代码, 证件类别代码, 证件号码, 派出所代码, 储户姓名, 开户日期, 储蓄类别代码, 储蓄类别, 采集日期, 采集方式)

用户组 (\*用户组代码, 用户组名称, 证明单出入库管理标志, 违规处理标志, 数据比对标志, 代码维护标志, 用户管理标志, 用户组管理标志, 数据查询标志, 数据统计标志)

违规处理 (\*违规类别代码, 违规类别, 处理结果代码, 处理时间, 用户代码)

处理结果 (\*处理结果代码, 处理结果描述, 有效标志)

证明单出入库 (\*流水号, 用户代码, 操作标志, 操作日期, 证明单始号, 证明单终号, 证明单份数, 领用税务机关代码, 发放税务机关代码, 学校代码, 领取人代码, 出入库标志, 有效标志)

证明单发放（\*流水号，学校代码，领用日期，发放始号，发放终号，发放数量，操作员代码，领取人姓名）

#### 4.2.3 证明单流水号功能的实现

证明单的管理需要设计唯一的序列号，称之为流水号。具体可以通过以下几种方式选择主键实现这个功能<sup>[45]</sup>。

##### （1）实现方法及其特点

第一种方法，以编号作主键。此方法就是采用实际业务中的唯一字段的“编号”作为主键设计，比如证明单领用审批表，会有“审批表编号”字段，而这个字段呢在业务实际中本身就是应该具有唯一性，具有唯一标识记录的功能，但它是包含业务逻辑具有实际意义的字段，数据库设计中原则上避免使用具有实际意义包含业务逻辑的字段作为主键。因为它具有无法预知的业务逻辑更改的可能性，比如要进行编号修改时，可能要涉及到很多相关联的其他表，这样原来的主键就面临危险了。因此，需新设一个字段专门用为主键，此主键本身在业务逻辑上不体现，不具有实际意义。

第二种方法，自动编号主键。就是新建一个 ID 字段，自动增长，满足主键的原则，优点是：数据库自动编号，速度快，而且是增量增长，聚集型主键按顺序存放，对于检索非常有利；数字型的，占用空间小，易排序，在程序中传递也方便；非常方便，不用担心主键重复问题。不象通过非系统增加记录（比如手动录入，或是用其他工具直接在表里插入新记录，或老系统数据导入）时，很可能产生主键重复问题。在 SQL Server 通过在表中设置 IDENTITY 列实现。缺点：因为自动增长，在手动要插入指定 ID 的记录时会显得麻烦，尤其是当系统与其他系统集成时，需要数据导入时，很难保证原系统的 ID 不发生主键冲突（前提是老系统也是数字型的）；如果其他系统主键不是数字型那就麻烦更大了，会导致修改主键数据类型了，这也会导致其他相关表的修改，后果同样很严重。

第三种方法，取最大值加一。采用自己生成的方法，同样是数字型的，只是把自动增长去掉了，采用在 Insert 时，读取 Max 值后加一，这种方法可以避免自动编号的问题，但也存在一个效率问题，如果记录非常大的话，那么 Max() 也会影响效率的；更严重的是并发性问题，如果同时有两人读到相同的 Max 后，加一

后插入的 ID 值会重复。

第四种方法,通过函数和存储过程自制加一的序列。考虑 Max 加一的效率后,采用自制加一,也就是建一个特别的表,字段为:表名,当前序列值。这样在往表中插入值时,先从此表中找到相应表的最大值后加一,进行插入,但可能会存在并发处理,这个并发处理,我们可以采用 Lock 线程的方式来避免,在生成此值的时,先 Lock,取到值以后,再 unLock 出来,这样不会有两人同时生成了。这比 Max 加一的速度要快多了。但同样存在一个问题:在与其他系统集成时,脱离了系统中的生成方法后,很难保证自制表中的最大值与导入后的保持一致。因此在“自制加一”中可以把主键设为字符型的,字符型主键可以应付很多我们意想不到的情况。

第五种方法, GUID 主键。目前一个比较好的主键是采用 GUID,主键可以是字符型的,但值由 GUID 生成, GUID 可以自动生成,也可以程序生成,而且键值不可能重复,可以解决系统集成问题,几个系统的 GUID 值导到一起时,也不会发生重复,就算有老数据也可以区分,而且效率很高,在 SQL 里设置 uniqueidentifier 列使用 NewID() 生成。优点是:同 IDENTITY 列相比,可以通过 NewID() 函数提前得知新增加的行 ID,为应用程序的后续处理提供了很大方便。便于数据库移植,其它数据库中并不一定具有 IDENTITY 列,而 GUID 列可以作为字符型列转换到其它数据库中,同时将应用程序中产生的 GUID 值存入数据库,它不会对原有数据带来影响。便于数据库初始化,如果应用程序要加载一些初始数据, IDENTITY 列的处理方式就比较麻烦,而 uniqueidentifier 列则无需任何处理,直接用 T-SQL 加载即可。便于对某些对象或常量进行永久标识。缺点是: GUID 值较长,不容易记忆和输入,而且这个值是随机、无顺序的 GUID 的值有 16 个字节,与其它那些诸如 4 字节的整数相比要相对大一些,存储空间增大,索引时间较慢,不适合需要自动增长有序的序列的应用。

综合考虑各种方法的特点,由于证明单的管理是属于教育储蓄个人所得税管理特有的,不存在数据导入的问题,并且需要自动增长的不重复的序列号,因此在系统中创建证明单发放表(xx\_xx\_zmdff)、证明单领用审批表(xx\_zmdcp)、证明单出入库表(xx\_zmdcrk)、证明单库存表(zmdkc)等多个表,表中设计了流水号(LSH)字段,作为主键约束。通过在 SQL Server 中设计 LSH 列 identity 属性自



动编号的方法来实现这一功能，可以非常方便的获得唯一的流水号，避免通过人工实现出现的重复问题。

## (2) IDENTITY 列定义

IDENTITY 列即标识列，这是一个自动增长列。每一个表中都可以设置一个标识列，它包含由系统产生的能够标识表中每一行数据的惟一序列值。将一个列作为表中的标识列，需要为该列定义 IDENTITY 属性，其格式如下：

IDENTITY[(seed,increment)]

seed 指初始值，即表中的第一行数据的标识列的取值，默认为 1；increment 指步长值，即每一个新标识值比上一个增长的幅度，其默认值也为 1。

为一个列定义 IDENTITY 属性后，每当向表中插入一个包括了标识列的行时，SQL Server 2000 会自动将当前最新的标识值加上步长值作为该行的标识值，插入到表中。IDENTITY 列可以有以下应用：

①引用 IDENTITY 列。IDENTITY 列可以在 SQL 查询中被引用，以此代替该列的列名。引用一个 IDENTITY 列，需要使用 IDENTITYCOL 关键字。例如：

```
USE jycx
GO
SELECT * FROM XX_ZMDFF
WHERE IDENTITYCOL=3
```

该语句的作用是在 jycx 数据库的 XX\_ZMDFF 表中找出标识列的值为 3 的记录，这与直接使用该列的列名进行查找的效果一样。

②获取标识列的初始值和步长值。通过使用函数 ident\_seed(表名)，用户可以获得表上标识列的 seed（初始）值；通过使用函数 (表名)，可以获得标识列的增量值（步长）。

例如：要查询 jycx 数据库中 XX\_ZMDFF 表的标识列的初始值和增量值，则可分别在查询分析器中输入下面的语句：

```
select ident_seed('XX_ZMDFF')
select ident_incr ('XX_ZMDFF')
```

③获取标识列的最新标识值。用户可以使用 @@IDENTITY 或者 SCOPE\_IDENTITY() 全局变量得到最新的标识值，例如要获取 jycx 数据库中

XX\_ZMDFF 表标识列的最新标识值，可用下面的语句：

```
select @@identity from XX_ZMDFF 或 select LSH = SCOPE_identity()
```

SCOPE\_IDENTITY()和@@IDENTITY 变量都是用来取得当前 session 中最后的 IDENTITY 值，关键的不同是 SCOPE\_IDENTITY ( ) 函数有作用范围，即 SCOPE\_IDENTITY()仅仅在当前代码范围内有效，相当于高级编程语言中的局部变量。而@@IDENTITY 的作用范围比 SCOPE\_IDENTITY()广。说明它们作用范围不同的一个简单的例子是如果 XX\_ZMDFF 表有触发器，而恰恰触发器中也有 insert 语句，那么 @@IDENTITY 变量就会被触发器中的 insert 语句改写，而 SCOPE\_IDENTITY()则不会。

我们在执行证明单发放或出入库的时候，需要涉及到多个表，证明单发放表 (xx\_zmdff)、证明单领用审批表 (xx\_zmdcp)、证明单出入库表(xx\_zmdcrk)、证明单库存表(zmdkc)，下面以证明单的发放为例进行说明。我们每执行一次证明单发放，我们就相应地在证明单库存表里减去发放的证明单数量，不能出现证明单发放表里添加了发放记录而库存表没有减少库存量这种情况，也不能出现证明单领用审批表没有插入成功而证明单发放表却插入成功这种情况，因为这些情况都不符合实际的业务逻辑或者会产生无效数据。通过事务处理就可很好解决这个问题，具体程序算法流程图如图 4-4 所示。

### 4.3 信息编码技术

在 MIS 中，信息编码是将事物或概念(编码对象)赋予有一定规律性、易于人和计算机识别与处理的符号。编码的过程是信息分类和标识的过程，科学的分类是根据编码对象的特征或属性，将信息按一定原则进行区分和归类，并排序生成唯一标识，以便管理和使用信息。企业的基础资源如人员、供应商、客户、物料、产品、会计科目等是客观存在的无序事物，只有进行编码后才能成为可处理的有序数据单元，形成企业的基础信息资源。另外，编码与分类对计算机来说十分重要，是进行信息分类、校核、汇总及检索的关键。企业的基本信息应科学编码、全局唯一，便于信息管理、共享和交换，可在所有并行分布式信息系统中进行流通。所以信息统一编码是企业实施信息化的基础，也是关系到信息化整体效果和成败的关键因素。

目前，为了迎接电子商务给全球带来的机遇和挑战，实现网络间信息的无缝集成和互操作性，世界标准化组织 ISO 进行了标准信息分类编码相关研究并制定

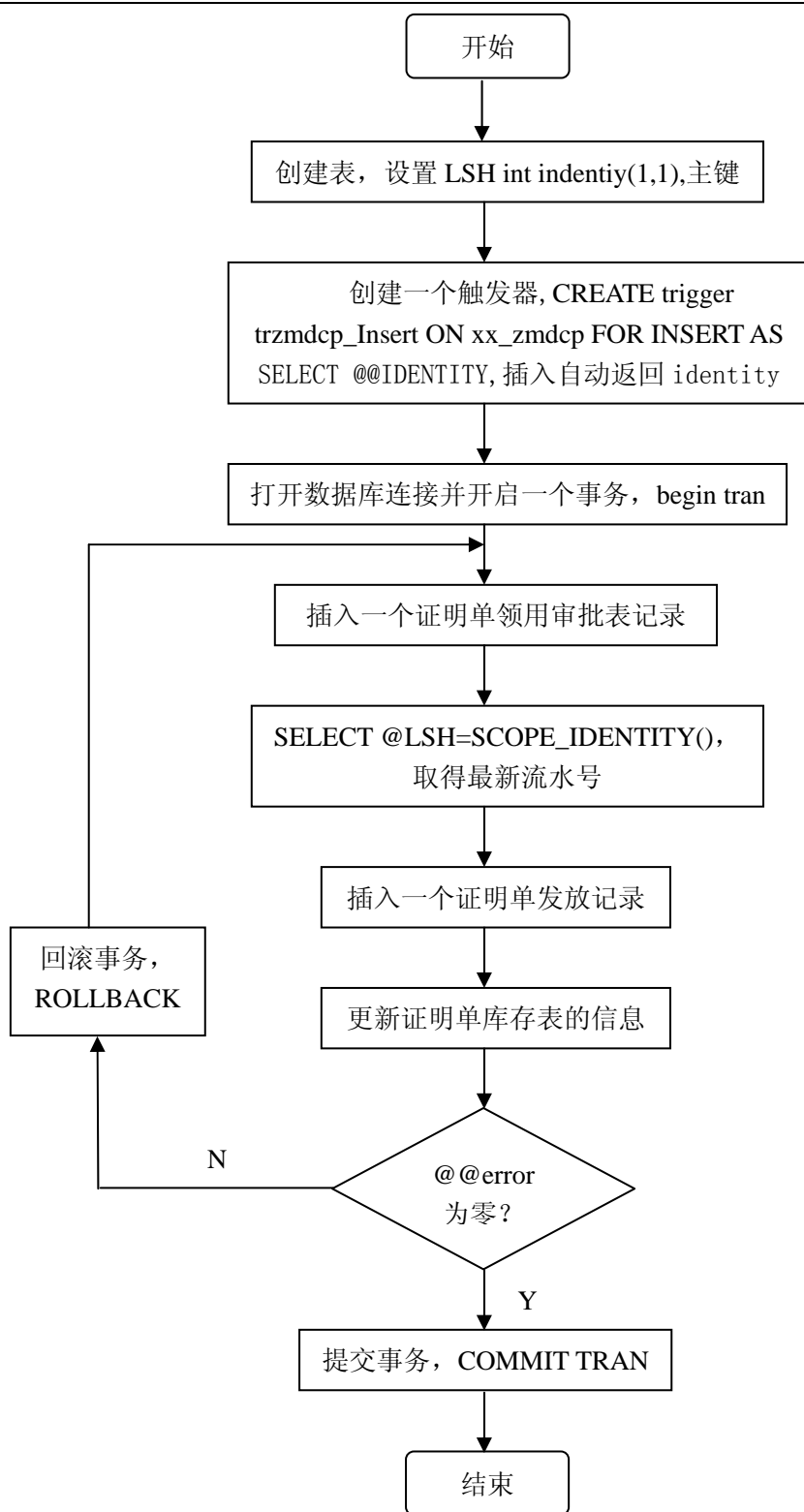


图 4-4 证明单发放的算法流程图

了有关标准。这些标准没有提供具体的分类方法，只为社会上流通的大宗物资提供编码隶属的范畴。在当前企业实施内部信息化过程中，这些标准并不实用和具有约束力，各企业根据自身需要确定编码和分类方法，难以适应供应链管理信息

流通的需要。在企业基础数据规范化方面存在着如下诸多带有普遍性的问题：各部门编码方法不统一或无分类规则，编码混乱；并行的信息系统编码方法不统一，无法进行信息的共享和系统集成；在管理上，无统一编码认证部门，存在严重的按名称流通等手工管理观念。

4.3.1 信息编码技术的概述

目前世界上有多种编码技术<sup>[16,17]</sup>，许多国家针对不同的应用需要已开发了不同的分类编码系统<sup>[18]</sup>。前西德阿亨工业大学的 OPITZ 教授在调查研究了 26 种机械产品、45000 种零件的基础上研制了 OPITZ 系统，该系统主要是由 5 位主码和 4 位辅码构成，属于以链式结构为主的混合式刚性码结构。主码中的前两位为树形结构，其他为链式结构，整个系统比较简单。但 OPITZ 系统为科学地应用成组技术提供了工具，开创了零件特征数字化的新纪元。

日本作为成组技术应用的后起之秀，吸取各国之长，首先提出了适合中小企业的 KC 分类编码系统<sup>[17]</sup>，紧接着又推出了适合大中型企业使用的 KK-1 ,KK -2 系统，随着成组技术应用的深入，又提出了 KK-3 分类编码系统。其中 KK-3 系统是一个多码位、多功用的通用分类编码系统，它由 21 位码构成。美国在许多高技术领域都应用了分类编码技术，如波音公司的 BUOCS 系统，它由 5 位主码和 7 位辅码构成，主要应用于企业设计、制造和管理。我国各部门也分别研制了本行业的分类编码系统，如航空部研制的 HFU 系统, 主要由 16 个码位构成，适用于航空附件企业的设计、制造和管理。

在进行信息编码时，应遵循唯一性、可扩充性、适应性、稳定性等基本原则，并且信息代码一般具有下列功能：识别功能、分类功能、排序功能、统计功能。

4.3.2 操作员编码的设计

在该系统中，操作员编码设置为 12 位，各位码所代表的意义，如表 4-1 所示。

表 4-1 操作员编码设置

编码意义	省行政区 码	地市行政 区码	县区行政 区码	税务分局 编码	科室号	操作员 号
代码位数(从左 到右)	1-2	3-4	5-6	7	8-9	10-12
位数	2 位	2 位	2 位	1 位	2 位	2 位

设置方法	系统初始设置	县级局统一编制
------	--------	---------

定位编码为 12 位分级组合码，全部采用数字编码，由县级税务机关按下述方法进行设置。前 6 位码按主体征管软件设置方法由本软件进行系统初始设置，为适应未来全省使用的需要增加了省行政区码。后 7-12 位码的设置分别为：

- (1)第 7 位码，用于在省辖市、县级市、县城区范围内，对应操作员所在的分局编码。
  - (2)第 8-9 位码，在操作员所在的分局范围内，对应操作员所在的科室。
  - (3)第 10-12 位码为操作员的顺序码。
- 根据该编码控制各县区操作员只能进行各县区局自己行政范围的操作和查询

4.4 OLE DB 技术

4.4.1 OLE DB 技术的概念

OLE DB 是 Microsoft 公司继 ODBC 之后提出的一套数据一致性访问接口，是 COM 技术在数据一致性访问方面的具体应用<sup>[45]</sup>。和 ODBC 相比，OLE DB 能够通过 SQL 命令访问关系型数据库，文件系统，电子邮件，甚至一个执行程序的输出结果。OLE DB 的实质是组件标准化技术在数据互操作性方面的应用。OLE DB 将传统的数据库系统划分为多个逻辑组件，这些组件之间相对独立又相互通信。它们的关系如图 4-5 所示。

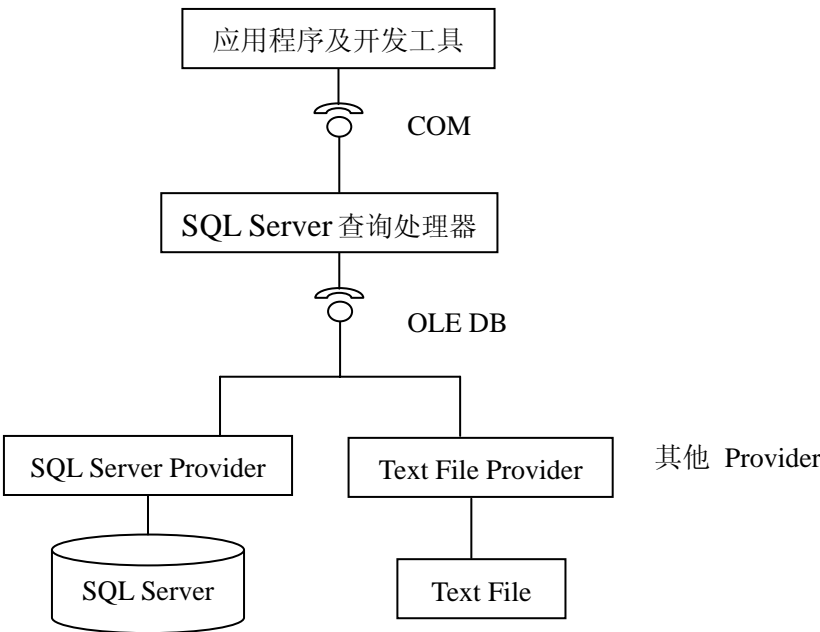


图 4-5 OLE DB 中逻辑组件关系图

这种组件模型中的各个部分被冠以不同的名称：

数据提供者（Data Provider）。提供数据存储的软件组件，小到普通的文本文件、大到主机上的复杂数据库都是数据提供者的例子。

数据服务提供者（Data Service Provider）。位于数据提供者之上、从过去的数据库管理系统中分离出来、独立运行的功能组件，例如查询处理器和游标引擎（Cursor Engine），这些组件使得数据提供者提供的数据以表状数据（Tabular Data）的形式向外表示（不管真实的物理数据是如何组织和存储的），并实现数据的查询和修改功能。SQL Server 的查询处理程序就是这种组件的典型例子。

业务组件（Business Component）。利用数据服务提供者、专门完成某种特定业务信息处理、可以重用的功能组件。分布式数据库应用系统中的中间层（Middle-Tier）就是这种组件的典型例子。

数据消费者（Data Consumer）。任何需要访问数据的系统程序或应用程序，除了典型的数据库应用程序之外，还包括需要访问各种数据源的开发工具或语言。

基于 COM 规范，OLE DB 定义了一批可扩展并且可以维护的接口，这些接口代管并封装 DBMS 功能中一致、可重复使用的部分。这些接口定义了 DBMS 组件的边界，例如行容器、查询处理器和事务处理协调器，使用这些组件可对各种信息源进行统一事务访问，从而把一个数据库组织成一个合作组件的基地。具体 OLE DB 是一套通过 COM 访问数据的 ActiveX 接口，按其功能分别包含在 Data Source、Session、Command 和 Rowset 等几个 COM 对象中，OLE DB 中主要对象的层次结构如图 4-6 所示。

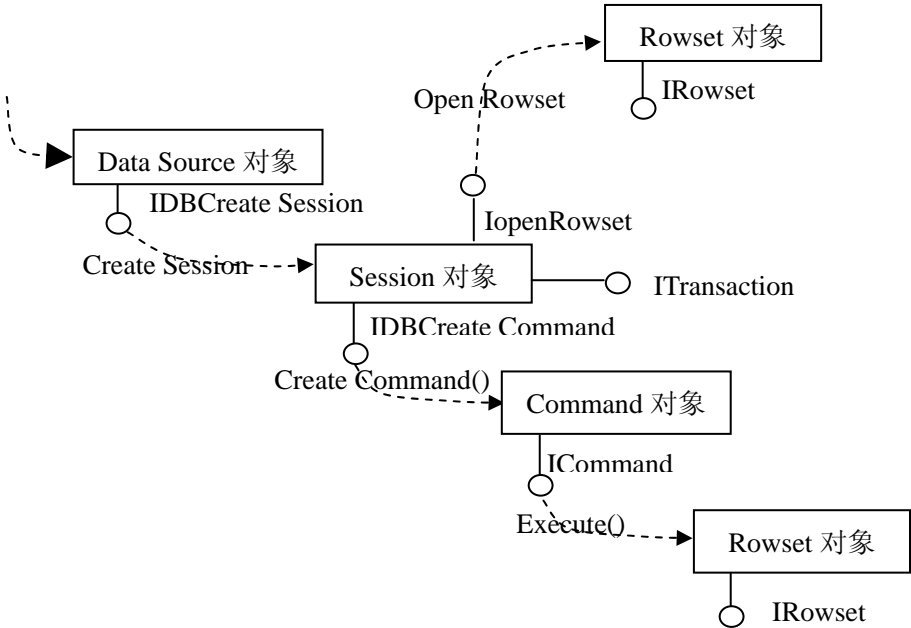


图 4-6 OLE DB 中主要对象的层次结构

Data Source 对象包含着和一个数据源的连接信息，通过实现各种不同的 Data Source 对象，就可以连接各种不同的数据源。

Session 对象代表和某个数据源的一次连接，一个 Data Source 对象可以产生多个 Session 对象，另外，Session 对象可以在一个本身不具有事务特性的数据源（例如文本文件）上，实现事务方式的操作。

Command 对象是用来执行命令的，例如 SQL 语句，针对不同的数据源，OLE DB Provider 实现的 Command 对象可以是 SQL 的，也可以是非 SQL 的（例如非关系型数据库），甚至可以不实现 Command 对象（例如对于文本文件数据源）。

Rowset 对象用表的格式表示数据，Rowset 对象是 OLE DB 中的核心对象，它使得 OLE DB 客户能够通过统一的表方式来访问和操作任何类型的数据源。例如对于文本文件中的数据，可以像关系型数据库中的表数据一样，通过 GetNextRows() 接口函数按行浏览数据。

#### 4.4.2 Microsoft 数据访问组件（MDAC）

通过 Microsoft 数据访问组件（MDAC），开发人员可以连接到种类繁多的关系和非关系数据源，并且使用这些数据源中的数据。您可以使用 ActiveX Data Objects (ADO)、开放式数据库连接 (ODBC) 或 OLE DB 连接到很多个不同的数据源。您可以通过由 Microsoft 生成和交付或者由各种第三方开发的提供程序和驱动程序完成该操作。另外，通过 MDAC（并根据不同的需求），开发的应用程序可以使用 ADO、OLE DB 或 ODBC 灵活地访问数据。MDAC 协议栈顶部是 API 或对象库层。应用程序通过对象库公开的 API 函数或接口连接到 Microsoft SQL Server。用于访问 SQL Server 的 API 示例包括 ODBC 和 DB-Library，现在除了没有其他组件可以使用时用一用，基本已经不使用。用于访问 SQL Server 的对象库示例包括 OLE DB、ADO 和 ADO.NET。由于 ADO 最终使用 OLE DB 与服务器通信，因此 Windows 应用程序在与 SQL Server 通信时实际上只使用两个常用的对象库，即 OLE DB 和 ADO.NET。由于通过 ADO 或 ADO.NET 进行连接通常比通过 ODBC 进行连接更普遍，如今，大多数应用程序均通过对象库（而非 ODBC 或类似 API）连接到 SQL Server。

该堆栈中的下一层是 Net-Library。Net-Library 在 API 或对象库（应用程序使用它与 SQL Server 进行通信）与网络协议（用于与网络交换数据）之间提供了一个

通道。SQL Server 为所有主要的网络协议提供了 Net-Library。这些库以透明方式将客户端发出的请求发送到 SQL Server，并将服务器发出的响应返回给客户端。可以使用 SQL Server 的客户端网络实用程序配置适用于特定客户端的 Net-Library。支持的客户端协议包括 TCP/IP、命名管道、NWLink、多协议 (RPC) 和其他一些协议。

当前 MDAC 版本支持的主要组件有：ADO，ADOMD，ADOX，OLE DB，SQLOLEDB，Microsoft SQL Server 网络库，ODBC，SQLODBC，开发新的应用程序或者升级现有的应用程序时可以选择使用这些组件。图 4-7 说明了可用于访问 SQL Server 2000 的不同技术之间的区别。

ADO 可以通过 ODBC 或 OLE DB 连接到 SQL Server 经 ODBC 连接是 ADO-->OLE DB-->ODBC Provider-->ODBC-->driver-->数据库；ADO 经 OLE DB 连接 是 ADO-->OLE DB-->DB Provider-->数据库；符合 ODBC 标准的数据源必须符合 OLE DB 标准，还必须提供相应的 OLE DB 服务程序。OLE DB 不通过任何附加技术层就可以直接访问 SQL Server，相比之下 OLE DB 连接数据库快，检索数据快，并且设置使用简便，所以我们选择了 OLE DB。

OLE DB 客户端(也称作使用者)通过客户端提供程序与服务器以及其他后端程序进行通信。此提供程序是一组 COM 组件(一个或多个)，用于将应用程序请求转换为网络进程间通信 (IPC) 请求。在使用 SQL Server 的情况下，最常用的 OLE DB 提供程序是 SQLOLEDB，它是 Microsoft 为 SQL Server 提供的 OLE DB 提供程序。SQLOLEDB 随附于 SQL Server 中，并作为 Microsoft 数据访问组件 (MDAC) 库的一部分安装。

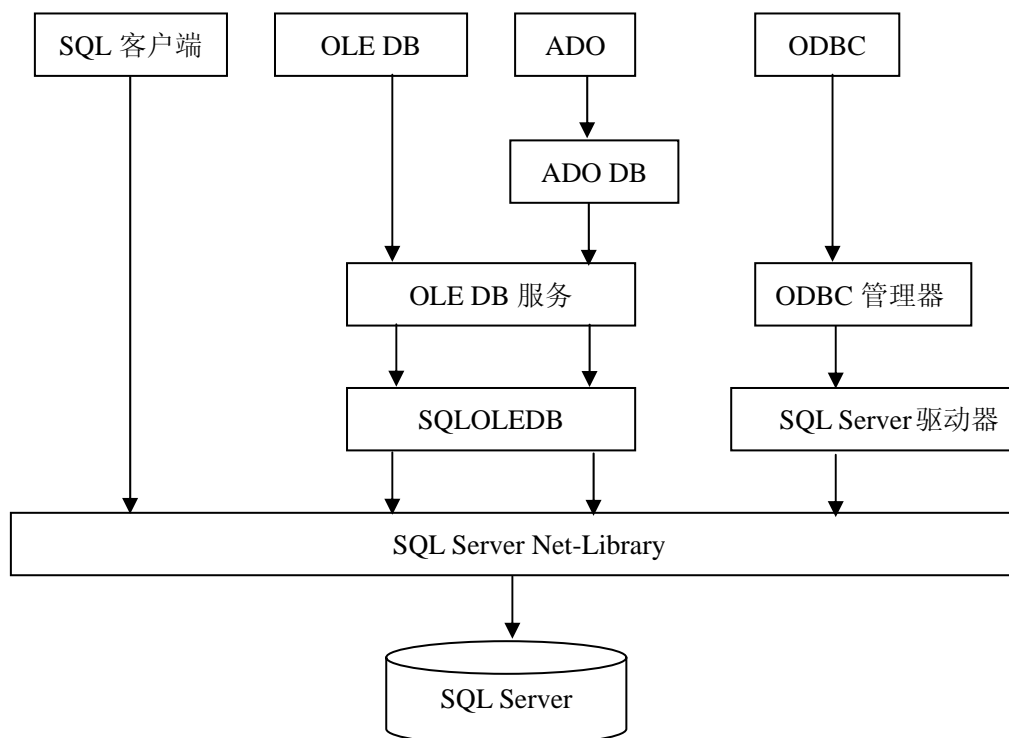




图 4-7 访问 SQL Server 2000 的连接方法

#### 4.4.3 OLE DB 在系统中的具体实现

在教育储蓄个人所得税管理系统中应用程序客户端采用 OLE DB 方式连接 SQL Server 2000 数据库, 具体分为以下设置步骤, 如图 4-8 所示:

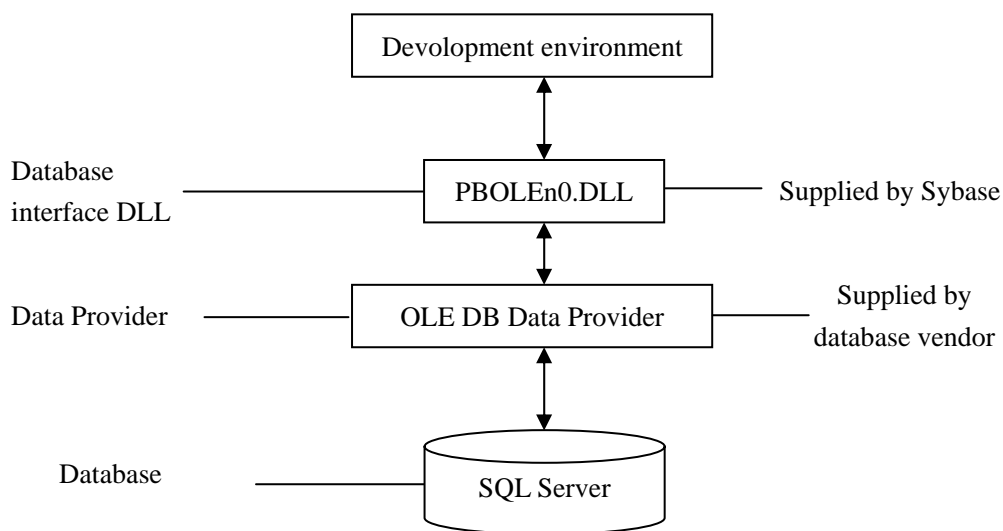


图 4-8 powerbuilder 与数据库的连接步骤

1) 服务器安装 SQL Server 并启动, 创建数据库, 配置网络, 安装配置客户端软件。网络采用 TCP/IP 协议。

2) 安装 OLE DB 数据库接口。在 PowerBuilder9.0 中提供了 SQL Server 2000 的 OLE DB 数据库接口。它使用动态链接库 PBOLExx.dll (xx 表示 PowerBuilder 的版本, 在这里是 PBOLE90.dll) 通过 OLE DB 连接 SQL Server 及其它数据库。在安装 PowerBuilder9.0 的时候选择 custom/full 方式安装 (默认的安装方式不会安装 OLE DB 接口)。

3) 客户机安装 SQL Server 客户端或者 MDAC 组件。Microsoft OLE DB Provider for SQL Server (SQLOLEDB.DLL) 随附于 SQL Server 中, 并作为 Microsoft 数据访问组件 (MDAC) 库的一部分安装。

4) 启动 PowerBuilder9.0, 配置数据源描述 (DB Profile), 通过直连接口 OLE DB, 定义 OLE DB 提供者。在数据库画板中找到 OLE microsoft OLE DB 项, 点击右键新建一个配置, 在弹出的配置窗口中:

profile name: 填写配置文件名, 如 jycx, 可任意取名。

provider: 选择数据库管理系统类型, 这里选择 “SQLOLEDB” 项。

**data source:** 服务器名。可以在下拉列表中选择,也可以填入服务器名或 IP 地址如: 85.48.18.102 等。如果上面的 **provider** 不是“SQLOLEDB”则此下拉列表中可能没有选项。

**user ID:** 用户名。这里填入 jycx。

**password:** 对应上面 user ID 的密码。

在 **Extended** 属性里填入 “**database=数据库名**”,例如 jycx。配置后进行连接测试,测试成功表示可以顺利的连接到 sql server 了。

也可以通过配置文件 (.ini) 实现数据库连接,如 jycx.ini 文件中数据库配置如下:

```
[database]
dbms="OLE DB"
autocommit=False
dbparm="DATASOURCE='85.48.18.102',PROVIDER='SQLOLEDB',PROVIDE
RSTRING='DATAbase=jycx',TIMEOUT=30"
```

在程序里创建数据窗口对象用 **profilestring** 读取。考虑到安全问题,数据库用户密码在程序中及后台进行处理。

## 4.5 系统性能优化

在进行客户机/服务器应用系统的开发时,生成具有较好性能的应用系统是十分重要的,如今的业务作业对计算机依赖的程度越来越高,应用系统的性能也越来越重要。我们发现许多的应用系统运行的速度相当慢,尤其是在进行大规模数据的统计计算和数据查询时。一个用 **Power Builder** 开发的 **MIS** 系统的性能包括处理最终用户查询的响应时间、一定时间内的吞吐量、数据完整性和安全级别以及日常运作的开销等。影响系统性能的因素很多,如硬件的配置、应用系统的设计及编程、网络方案、操作系统、网络和数据库的性能都可能成为影响应用系统性能的瓶颈。

在我们所开发的教育储蓄个人所得税管理系统中,服务器主频为 2.6GHZ,内存 512M,主干网采用 100M 快速交换式以太网,开发平台采用运行于 Windows 2000 的 Microsoft SQL SERVER 2000,客户端应用采用 Power Builder 9.0 开发,下面从系统体系结构、存储过程、视图和利用数据缓存几方面来分析该系统的性能优化过程。

### 1、该系统采用客户/服务器体系结构

#### (1) 客户、服务器工作负载的平衡

一般情况下，服务器最适合处理基于集合的数据检索和修改操作。客户工作站在显示复杂的用户界面，处理数据格式化和特殊列或行的数据有效验证时效率最高。设计时应注意充分利用它们各自的特点，发挥优势，均衡负载。

### (2) 减少网络负载

网络是客户/服务器工作的基础，网络的有限带宽也使它常常成为一个主要的瓶颈。减少网络负载，可以改善系统的响应时间，吞吐量。使用存储过程而不是进行大量的 SQL 查询可以有效地减少网络流量。这样，客户仅仅需要传递一些参数，对存储过程进行调用，而不是向服务器发送大量的 SQL 语句。另一方面，充分利用服务器的优势，由服务器来进行处理；而不是把原始数据传递到工作站再进行处理。可以减小结果集合，从而减小网络上传输的数据量。

### (3) 强大的数据操纵和事务处理能力

C/S 模式可实现分布的处理与集中的数据管理相结合，数据安全性强以及数据完整性约束，通信速度快，具有强大的数据操纵和事务处理能力。

## 2、使用存储过程

存储过程 (Stored Procedure) 是为了完成特定的功能而汇集成一组的 SQL 语句集，并为该组 SQL 语句命名，经编译后存储在 SQL Server 的数据库中。用户可以根据需要决定是否在每次执行时让 SQL Server 进行重新编译。用户可以指定存储过程的名字和给出参数来执行它。存储过程在第一次执行时建立优化的查询方案，SQL Server 将查询方案保存在高速缓存中，在接下来的运行中就可以直接从高速缓存执行。省去了优化和编译阶段，从而节省了执行所需的时间<sup>[30,32]</sup>。

最优的查询方案，往往要根据实际的要求和具体情况，通过比较进行选择。SQL Server 提供的功能可以对不同的查询结构的性能进行比较，包括查询计划、索引选择、I/O 次数、响应时间等。在开发过程中，可以使用这一有利工具。

客户/服务器方式下的应用程序中，客户机应用通常包括多个 SQL 语句进程，用以完成相应的数据检索、数据更新等任务。如果每个数据库请求均由应用程序以单个 SQL 语句的形式发送给服务器，则客户机和服务器之间的数据传送量显然大大增加。同时，数据库系统对 SQL 语句的语法检查、编译、分析和优化工作量也是相当大的。SQL 语句功能可以将多个 SQL 语句集合在一起形成存储过程。存储过程是存储数据库里的以 SQL 编写的代码段，是独立存在于表之外的数据库对象。存储过程在首次运行时，服务器必须对其进行优化和编译，编译后的优化方案存储在过程 cache 中。这样，在以后运行该过程时就可以充分利用此结果，跳过优化及编译过程，只要通过传输存储过程名及其参数 DBMS 就可以执行，这样就大大减少了数据请求和数据响应次数。同时，存储过程与其他

应用程序共享应用程序的逻辑，从而确保对数据访问和操纵的一致性，当业务需求发生改变时，只要改动存储过程，对客户端的影响可以减低到最低。存储过程提供的事务处理功能为大量数据处理的安全性和有效性提供了保证。因此，使用存储过程能极大的提高 SQL SERVER 的功能、效率、灵活性和安全性，并大大改善 SQL 语句和批处理的性能。

在数据库的开发过程中，经常会遇到复杂的业务逻辑和对数据库的操作，这个时候就会用存储过程来封装数据库操作。如果项目的存储过程较多，书写又没有一定的规范，将会影响以后的系统维护困难和大存储过程逻辑的难以理解，另外如果数据库的数据量大或者项目对存储过程的性能要求很高，就会遇到优化的问题，否则速度有可能很慢，一个经过优化过的存储过程要比一个性能差的存储过程的效率甚至高几百倍在 C/S 开发中不可避免地要接触到对后台数据的大量处理。在教育储蓄个人所得税管理系统中的比对过程中，牵涉的表格和数据量较大，在编写存储过程中反复对 SQL 语句进行优化，使比对时间达到用户需求。可见以于一个系统不是简单地能实现其功能就可，而是要写出高质量的 SQL 语句，提高系统的可用性，优化总结如下<sup>[13]</sup>：

#### 1. IS NULL 与 IS NOT NULL

不能用 null 作索引，任何包含 null 值的列都不会被包含在索引中。即使索引有多列这样的情况下，只要这些列中有一列含有 null，该列就会从索引中排除。也就是说如果某列存在空值，即使对该列建索引也不会提高性能。任何在 where 子句中使用 is null 或 is not null 的语句优化器是不允许使用索引的。

#### 2. 带通配符 (%) 的 like 语句

目前的需求是这样的，要求在银行代码表中查询银行名称中包含“工商银行”的银行。可以采用如下的查询 SQL 语句：

```
select * from DM_YH where YHMC like '%工商银行%';
```

这里由于通配符(%)在搜寻词首出现，所以 SQL Server 系统不使用银行名称的索引。在很多情况下可能无法避免这种情况，但是一定要心中有数，通配符如此使用会降低查询速度。然而当通配符出现在字符串其他位置时，优化器就能利用索引。在下面的查询中索引得到了使用：

```
select * from YHMC where last_name like '工%';
```

#### 3. Order by 语句

Order by 语句决定了 Oracle 如何将返回的查询结果排序。LSH by 语句对要排序的列没有什么特别的限制，也可以将函数加入列中（象联接或者附加等）。任何在 LSH by 语句的非索引项或者有计算表达式都将降低查询速度。仔细检查 LSH by 语句以找出非索引项或者表达式，它们会降低性能。解决这个问题的办

法就是重写 **LSH by** 语句以使用索引，也可以为所使用的列建立另外一个索引，同时应绝对避免在 **LSH by** 子句中使用表达式。

#### 4. NOT

我们在查询时经常在 **where** 子句使用一些逻辑表达式，如大于、小于、等于以及不等于等等，也可以使用 **and**（与）、**or**（或）以及 **not**（非）。**NOT** 可用来对任何逻辑运算符取反。下面是一个 **NOT** 子句的例子：

```
... where not (status = 'VALID')
```

如果要使用 **NOT**，则应在取反的短语前面加上括号，并在短语前面加上 **NOT** 运算符。**NOT** 运算符包含在另外一个逻辑运算符中，这就是不等于（<>）运算符。换句话说，即使不在查询 **where** 子句中显式地加入 **NOT** 词，**NOT** 仍在运算符中，见下例：

```
... where status <> 'INVALID';
```

再看下面这个例子：

```
select * from XX_CKQK where JE<>20000;
```

对这个查询，可以改写为不使用 **NOT**：

```
select * from XX_CKQK where JE<20000 or JE>20000;
```

虽然这两种查询的结果一样，但是第二种查询方案会比第一种查询方案更快些。第二种查询允许 **SQL Server** 对 **JE** 列使用索引，而第一种查询则不能使用索引。

#### 5. IN 和 EXISTS

有时候会将一列和一系列值相比较。最简单的办法就是在 **where** 子句中使用子查询。在 **where** 子句中可以使用两种格式的子查询。

第一种格式是使用 **IN** 操作符：

```
... where column in(select * from ... where ...);
```

第二种格式是使用 **EXIST** 操作符：

```
... where exists (select 'X' from ...where ...);
```

我相信绝大多数人会使用第一种格式，因为它比较容易编写，而实际上第二种格式要远比第一种格式的效率。在 **SQL Server** 中可以几乎将所有的 **IN** 操作符子查询改写为使用 **EXISTS** 的子查询。

第二种格式中，子查询以 ‘select 'X' 开始。运用 **EXISTS** 子句不管子查询从表中抽取什么数据它只查看 **where** 子句。这样优化器就不必遍历整个表而仅根据索引就可完成工作（这里假定在 **where** 语句中使用的列存在索引）。相对于 **IN** 子句来说，**EXISTS** 使用相连子查询，构造起来要比 **IN** 子查询困难一些。通过使用 **EXIST**，**SQL Server** 系统会首先检查主查询，然后运行子查询直到它

找到第一个匹配项，这就节省了时间。SQL Server 系统在执行 IN 子查询时，首先执行子查询，并将获得的结果列表存放在在一个加了索引的临时表中。在执行子查询之前，系统先将主查询挂起，待子查询执行完毕，存放在临时表中以后再执行主查询。这也就是使用 EXISTS 比使用 IN 通常查询速度快的原因。同时应尽可能使用 NOT EXISTS 来代替 NOT IN，尽管二者都使用了 NOT（不能使用索引而降低速度），NOT EXISTS 要比 NOT IN 查询效率更高。

例如教育储蓄个人所得税管理系统的比对过程，牵涉到的表格和数据量极大，时间开销较大，这时我们就要充分利用 SQL Server 2000 提供的存储过程功能来实现，下面是系统中数据比对存储过程中进行优化后的 SQL 语句，经过优化后时间开销由数分钟提高到数十秒。随着应用系统的实际应用，随着数据库中数据的增加，更能体现其性能的优劣。

```
--BEGIN TRANSACTION
--存款比对
print '正在进行存款比对.....'
--一次存款额超过 2 万的帐户
insert jg_dyyhwg
select a.yh_dm, a.yhzh, @today, '05', NULL, NULL, NULL
from xx_ch a, cl_dyckqk b
where a.yx_bz='1' and a.yh_dm=b.yh_dm and a.yhzh=b.yhzh and b.cq_bz='0' and
b.je>=20000
if @@error<>0
begin
raiserror 2700421 '存款比对处理-1 失败，比对退出!'
--select @int_bz = 2
ROLLBACK TRANSACTION
return
end
--累计存款额超过 2 万的帐户：
--证件类型为 '0' 即无证件的帐户比较
insert jg_dyyhwg
select yh_dm, yhzh, @today, '06', NULL, NULL, NULL
from xx_ch
where yx_bz='1' and zjlz='0' and ckze>20000
if @@error<>0
```

```
begin
raiserror 2700422 '存款比对处理-2 失败, 比对退出!'
--select @int_bz = 2
ROLLBACK TRANSACTION
return
end
--证件类型为 '2' 即有身份证号的帐户比较
insert jg_dyhwg
select a.yh_dm, a.yhzh, @today, '06', NULL, NULL, NULL
from xx_ch a
where a.yx_bz='1' and a.zjlb= '2' and exists
(select b.zjhm from xx_ch b
where b.yx_bz='1' and b.zjlb= '2' and b.zjhm=a.zjhm
group by b.zjhm
having sum(b.ckze)>20000
)
if @@error<>0
begin
raiserror 2700423 '存款比对处理-3 失败, 比对退出!'
--select @int_bz = 2
ROLLBACK TRANSACTION
return
end
--证件类型为 '1' 即有户口本的帐户比较
insert jg_dyhwg
select a.yh_dm, a.yhzh, @today, '06', NULL, NULL, NULL
from xx_ch a
where a.yx_bz='1' and a.zjlb= '1' and exists
(select b.zjhm, b.chxm from xx_ch b
where b.yx_bz='1' and b.zjlb= '1' and b.zjhm=a.zjhm and b.chxm=a.chxm
group by b.zjhm, b.chxm
having sum(b.ckze)>20000
)
if @@error<>0
```

```
begin
raiserror 2700424 '存款比对处理-4 失败, 比对退出!'
--select @int_bz = 2
ROLLBACK TRANSACTION
return
end
--COMMIT TRANSACTION
print '存款比对处理成功!'
--return
```

存储过程的使用非常方便。它的最大优点就是可提供多个用户使用,从而减少 PowerBuilder 应用程序开发中的冗余劳动。对 SQL 语句的优化大大减少对程序逻辑的改变,避免源代码优化的高时间成本和高风险代价。

### 3、视图

视图是一个存储的 SELECT 语句,每当在 SQL 语句中使用该视图名,就执行这个 SELECT 语句。视图并不保留数据的拷贝,因此通过视图看到的数据总是最新的。如果视图是个简单视图(比如不包括概括数据,底层查询的 SELECT 语句不修改列,不做表连接),则可以通过视图进行插入、更新和删除操作。

系统开发中在数据库后台的各种数据更新做到了全程事务控制,即要么所有操作全部正确更新完毕,要么就全部滚回,保证了数据的有效性,同时对采集数据采取了原封不动地存档后才进行处理的措施,这样,一旦有操作失误,因为有原始数据的存在,整个数据库可有效地进行恢复,避免了一次失误导致整个数据库数据不可用的风险。

同时我们充分利用 SQL2000 中的视图来简化对复杂数据的访问,简化 PowerBuild 客户端的开发,对常用多表关联查询建立视图,PowerBuild 客户端开发者可以把视图当作一个表来处理,对于业务需求变更时可以极大地节省客户端的修改。使用视图对应用程序表中的数据进行及时概括,这样通过视图上的简单查询就可以得到复杂的概括数据,即使表中的数据改变了,视图也总能得出最新的信息。下面是教育储蓄个人所得税管理系统所创建 VIEW\_JYCX\_DYXXWG 视图的脚本:

```
CREATE VIEW dbo.VIEW_JYCX_DYXXWG
AS
SELECT T1.ZMD_BH AS 证明单编号, T1.XX_DM AS 学校代码, T2.XXMC
AS 学校名称, T1.YH_DM AS 银行代码, T3.YHMC AS 银行名称, T1.YHZH AS
银行帐号,
```



T5.ZJMS AS 证件类别, T6.PCSMC AS 派出所名称, T4.ZJHM AS 证件号码, T4.CHXM AS 储户姓名, T1.BDRQ AS 比对日期, T1.WG\_DM AS 违规代码, T7.WGMS AS 违规描述, T1.CLRQ AS 处理日期, T1.CLJG\_DM AS 处理结果代码, T1.CZY\_DM AS 处理人代码

```
FROM dbo.jg_dyxxwg T1 INNER JOIN
    dbo.DM_XX T2 ON T1.XX_DM = T2.XX_DM INNER JOIN
    dbo.DM_YH T3 ON T1.YH_DM = T3.YH_DM INNER JOIN
    dbo.XX_CH T4 ON T1.YH_DM = T4.YH_DM AND T1.YHZH =
T4.YHZH INNER JOIN
    dbo.DM_ZJLB T5 ON T4.ZJLB = T5.ZJLB INNER JOIN
    dbo.DM_PCS T6 ON T4.PCS_DM = T6.PCS_DM INNER JOIN
    dbo.DM_WGLB T7 ON T1.WG_DM = T7.WG_DM
```

4、利用数据缓存：在 Power Builder 中提供了一种易用而高效的信息访问方式——Data Window (数据窗口)大量的数据检索一般都是通过数据窗口完成的，因此充分利用数据窗口缓存区(包括主缓存区，筛选缓存区，删除缓存区)中现有的数据集，客户机应为那些使用频繁，保持不变的数据开辟数据缓冲区，以避免多次的网络数据检索。

应用程序的代码设计也要进行必要的优化，以提高应用运行的速度。一段糟糕的代码会大大影响系统最后的运行效果。

系统性能的优化贯穿于系统的整个生命周期，从设计到开发到维护，其中的每个步骤，哪怕是很小的不经意，都会对最终的性能产生一定的影响。优化意识应当始终受到关注。优化策略的实施必须依赖于用户的软、硬件配置，应用程序设计和实施，以及最终的作用和性能要求。对网络瓶颈、存储结构、事务特点、性能要求的深入分析和理解是优化的基础。另外也应该认识到，提高性能的各种策略都有自己的利弊，一种策略的实施可能使一些方面受益，而损害其它方面的性能。在实际系统中，必须进行适当的折衷和平衡。

## 4.6 系统安全技术

在该系统中，存储了大量重要的、敏感的数据，它涉及到银行、储户的保密要求，涉及到国家和纳税人的利益，在系统运行过程中，业务管理人员和用户的操作都要涉及到这些敏感数据的处理。因此，该系统在安全性方面有一定的要求。而对于一个系统来说，构建一个完整的安全机制应包括基于数据库的安全、基于应用系统的安全和基于网络的安全。我们在系统的设计与开发中主要采用下列安全技术，从而构建一个比较全面的安全保护机制，如图 4-9 所示。

4.6.1 数据库安全技术

(1) 安全管理

在设计的时候，根据具体情况，设立专门的管理员负责数据库结构的维护、系统人员管理等工作。其他用户只能直接使用数据库中的数据，但是不能对数据库结构进行更改，也不能了解其他用户的信息。

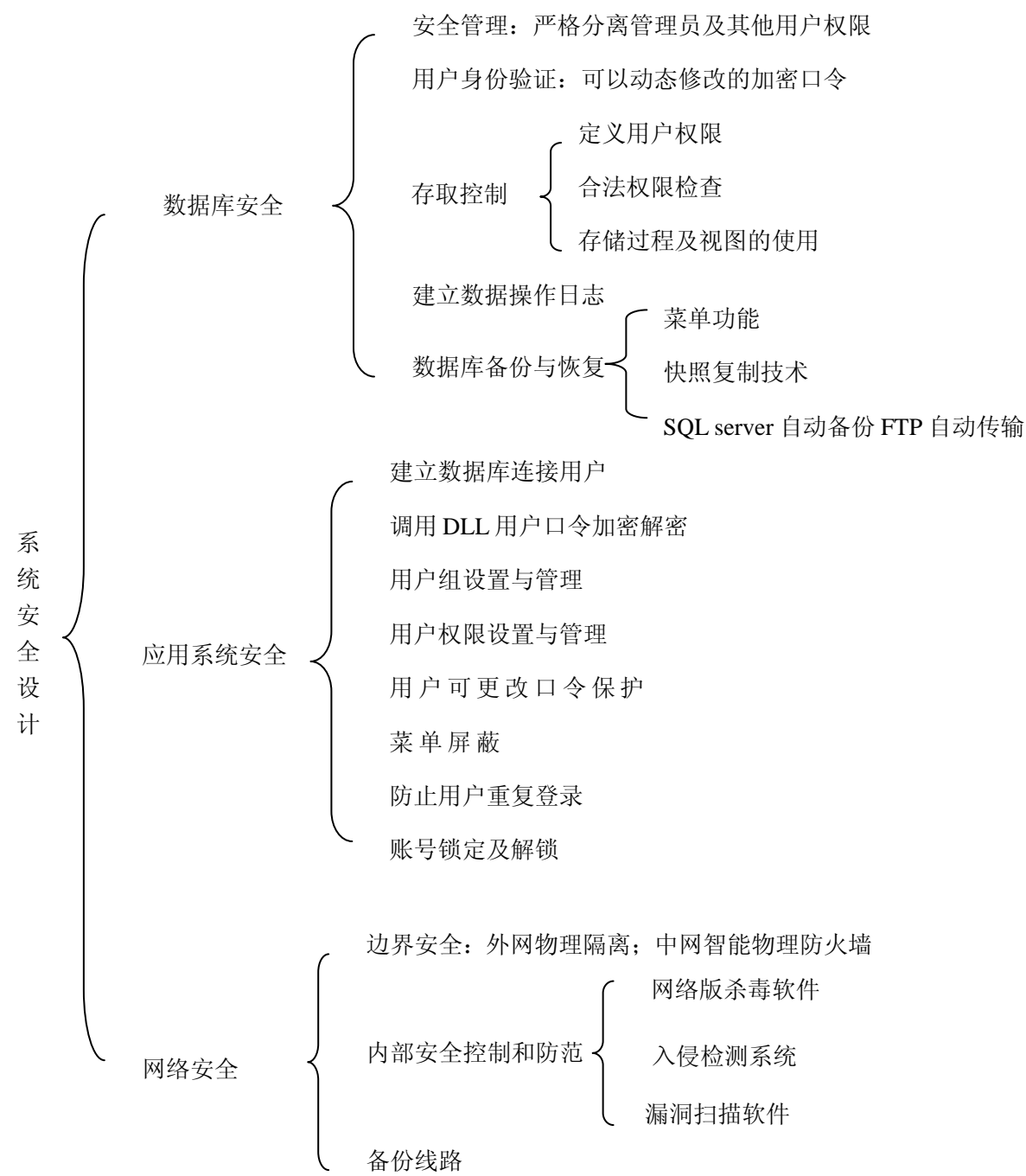


图 4-9 系统数据安全保护机制

### (2) 用户身份验证

该系统中包含大量的敏感数据，为保证系统数据在存储时和网络传输中不被未经授权的用户访问或解读，所以在用户进入信息系统时，对用户进行身份验证。利用 SQL Server 提供的安全认证创建相应的数据库帐户，任何用户对数据库的任何操作都必须强制通过系统的安全设置检查之后方能实现。

首先，设置运行口令核实用户身份。口令值存放在某一单独的数据库表中，用户可以修改口令。整个系统的安全在一定程度上基于口令的保密，为保证口令的安全性，对口令的长度、更换口令的最短期限等作出明确规定。口令在传输时必须加密。存放口令的数据库文件是经过加密处理的，密钥是一个函数  $f$ ，若用户输入口令  $x$ ，则口令的库文件中实际存放的是  $f(x)$ ，这样就可以保证其安全性。

其次，用户按工作岗位分组，即利用 SQL Server 角色进行统一管理，严禁混岗现象发生。属于同一工作岗位的用户具有相同功能的权限，SQL Server 中将访问许可集中授予角色，每个组作为一个角色加强管理用户的访问许可。角色（工作组）建立后，为各组添加用户。用户与角色可以建立多值的对应关系。一个用户可以对应多种角色，同一角色也可以对应多个用户。这种多值关系可以满足用户多种权限设置。权限设置由系统管理员统一管理。具体操作权限的设置由存取控制来完成。

### (3) 存取控制

口令的作用只是用来鉴别用户，它是防止入侵者侵入系统的第一道防线，而不是系统内控制主体对客体访问的措施。存取控制是从计算机系统的处理功能方面对数据提供保护，它是数据保护的前沿屏障。数据库安全性的基本原理是控制用户对数据库的访问。只有经识别的被允许的人才具有检查、生成、删除和修改信息的权利。同时令所有未被授权的人员无法接近数据。这是通过数据库系统的存取控制机制实现的。它包括两部分：

①定义用户权限，并将用户权限登记到数据字典中。用户权限是指不同用户对不同数据对象允许执行的操作权限。用户权限被定义并经过编译后存放在数据字典中，即作为系统的安全规则或授权规则。

②合法权限检查。当用户发出存取数据库的操作请求后，数据库管理系统查找数据字典，根据安全规则进行合法权限检查，若用户操作请求超出了定义权限，系统将拒绝执行此操作。

③存储过程及视图的使用。一些必须允许用户修改的敏感的表，并且在很少的情况下，这时创建一个存储过程来检查是否满足该表特殊的要求，然后再执行插入式更新操作。普通用户对这个敏感表不具有插入和更新权限，可以将该存储过程作为一个具有插入和更新权限的用户，然后将执行该存储过程的权限授予那

些用户。在更新该表之前由这个存储过程检查用户的操作是否合法。使用视图可以提供数据库管理系统标准软件所不能提供的安全性。如果要将一个表上的某些权限赋予另外一个数据库用户，但又想限制他访问特定行和特定列的数据，这时可以创建一个视图，然后将该视图上的权限授予他。

访问权限是以数据库为对象而设置的，被授予数据库用户。每一位用户对不同的数据库都拥有不同级别的访问权限。

(4) 建立数据操作日志

对进行关键性操作的用户的用户名、权限级别、进入时间、退出时间、执行的操作及操作对象进行记录, 以便事后进行责任跟踪, 同时也为日后的操作量、录入数据质量的考核提供依据。

(5) 数据库的数据备份与恢复

数据库的备份是恢复数据最容易和最有效的保证方法之一，当服务器发生故障时，利用备份进行数据库恢复，以恢复破坏的数据库文件、控制文件或其他文件。在该系统中通过三种方式实现数据库备份和恢复的<sup>[29]</sup>。

第一种方式是在系统维护中设置了备份和恢复功能，通过存储过程来实现数据库的本地备份，用于系统管理员平时系统维护，使用方便快捷。

第二种方式是使用SQL Server 2000的数据复制技术中的快照复制将数据库复制到网络中已经安装SQL Server 2000的服务器上，通过非工作时间的发布和订阅，在不影响性能的同时实现数据库同步，以便能很快接替故障服务器的工作。

SQL Server 有三种复制技术，分别为快照复制、事务复制和合并复制。快照复制<sup>[38]</sup>是完全按照数据和数据库对象出现时的状态对其进行复制和分发的过程。这种复制方式一次即复制整个数据集，因此不要求对更改进行连续的监视，数据修改传播到订阅服务器的时间要更长一些，而且为防止复制占用过多资源，复制快照发布的频率通常低于其它发布类型。因此，快照复制适用于所复制数据较为静态、规模较小、且可接受复制的高滞后时间的情况。在复制环境中，根据复制数据的链路方向不同，将服务器分为发布服务器、分发服务器和订阅服务器。快照模式下数据源复制工作示意图如图 4-10 所示。

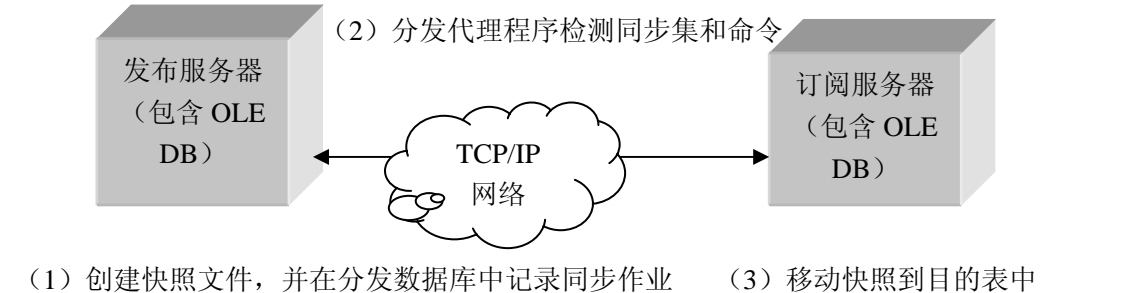


图 4-10 快照模式下数据源复制工作示意图

快照复制是通过快照代理程序和分发代理程序实现的。快照代理程序每次运行时，都会检查是否新增了任何新订阅。如果创建发布时启用了“立即创建第一个快照”选项，则每次快照代理程序运行时都会创建新的快照文件，其中包含了已发布表和数据库对象的架构和数据，然后将这些文件存储在快照文件夹中，并在分发数据库中记录同步作业。分发代理程序定期检查分发数据库中的同步集的位置和订阅服务器同步命令，并将保存在分发数据库表中的快照移动到订阅服务器上的目的表中。在教育储蓄个人所得税管理系统服务器上建立 jycx 数据库的快照复制项目，代码如下。

--启用复制数据库

```
use master
```

```
GO
```

```
exec sp_replicationdboption @dbname = N'jycx', @optname = N'publish', @value  
= N'true'
```

```
GO
```

```
use [jycx]
```

```
GO
```

---添加快照发布

```
exec sp_addpublication @publication = N'jycx', @restricted = N'false',  
@sync_method = N'character', @repl_freq = N'snapshot', @description = N'来  
自发布服务器的 jycx 数据库的快照发布。', @status =
```

```
N'active', @allow_push = N'true', @allow_pull = N'true', @allow_anonymous =  
N'false', @enabled_for_internet = N'false', @independent_agent = N'false',
```

```
@immediate_sync = N'false', @allow_sync_tran = N'false',
```

```
@autogen_sync_procs = N'false', @retention = 336, @allow_queued_tran =
```

```
N'false', @snapshot_in_defaultfolder = N'true', @compress_snapshot = N'false',
```

```
@ftp_port = 21, @ftp_login = N'anonymous', @allow_dts = N'false',
```

```
@allow_subscription_copy = N'false', @add_to_active_directory = N'false'
```

```
exec sp_addpublication_snapshot @publication = N'jycx', @frequency_type = 4,
```

```
@frequency_interval = 1, @frequency_relative_interval = 0,
```

```
@frequency_recurrence_factor = 1, @frequency_subday = 1,
```

```
@frequency_subday_interval = 0, @active_start_date = 0, @active_end_date =
```

```
0, @active_start_time_of_day = 220300, @active_end_time_of_day = 0,
```

```
@snapshot_job_name = N'85.48.16.109-jycx-jycx-1'
```

```
GO
```

```
exec sp_grant_publication_access @publication = N'jycx', @login =  
N'BUILTIN\Administrators'  
GO  
exec sp_grant_publication_access @publication = N'jycx', @login =  
N'distributor_admin'  
GO  
exec sp_grant_publication_access @publication = N'jycx', @login = N'sa'  
GO  
---添加快照项目  
exec sp_addarticle @publication = N'jycx', @article = N'dm_swjg',  
@source_owner = N'dbo', @source_object = N' dm_swjg ', @destination_table =  
N'dm_swjg', @type = N'logbased', @creation_script = null, @description = null,  
@pre_creation_cmd = N'drop', @schema_option = 0x00000000000000F1,  
@status = 0, @vertical_partition = N'false', @ins_cmd = N'SQL', @del_cmd =  
N'SQL', @upd_cmd = N'SQL', @filter = null, @sync_object = null,  
@auto_identity_range = N'false'  
GO  
exec sp_addarticle @publication = N'jycx', @article = N'xx_zmd', @source_owner  
= N'dbo', @source_object = N' xx_zmd ', @destination_table = N'xx_zmd ', @type  
= N'logbased', @creation_script = null, @description = null, @pre_creation_cmd  
= N'drop', @schema_option = 0x00000000000000F1, @status = 0,  
@vertical_partition = N'false', @ins_cmd = N'SQL', @del_cmd = N'SQL',  
@upd_cmd = N'SQL', @filter = null, @sync_object = null,  
@auto_identity_range = N'false'.....  
GO
```

简要说明以上代码的工作机制：上述代码首先打开 master 系统数据库，执行 sp\_replicationdboption 系统存储过程，启用对当前数据库 jycx 的发布。在打开当前数据库 jycx 后，通过执行 sp\_addpublication、sp\_addpulation\_snapshot 和 sp\_grant\_publication\_access 系统存储过程来创建对服务器上 jycx 数据库的快照发布。其中，sp\_addpulation 存储过程定义发布选项，如：发布模式 repl\_freq、同步方式 sync\_method 和发布描述 description 等发布定义选项；sp\_addpublication\_snapshot 存储过程创建快照代理程序，设置发布 agent\_id，并在复制工作目录中放入架构和数据；sp\_grant\_publication\_access 存储过程用于向发布的访问列表中添加登陆。最后通过重复执行 sp\_addarticle 存储过程完成对快照发布

中各项目的定义。

第三种方式是通过 SQL Server 的数据库备份和恢复功能,采用主用机和备用机备份自动FTP传输方式进行异地备份。该方式利用MSDE内部的SQL Server2000 agent 的 JOB 完成数据的自动备份并上传到 FTP 服务器,完成数据的自动备份。首先通过数据库维护计划实现自动备份,再通过 SQL Server2000 agent server 定义作业。下面说明系统中具体实施方法:

①在安装系统的主用服务器上的用来备份的分区的根目录下建立 SJBF 目录,用于存放备份的教育储蓄个人所得税管理系统的数据库文件。

②在主用服务器上打开企业管理器（Enterprise Manager），点击开始—>Microsoft SQL Server2000—> Enterprise Manager。在注册本机连接后在数据维护计划中设置自动备份，按提示选择数据库，进行相关选项和调度的设置，备份目录设置为第一步建立的目录。如图 4-11 所示：

③在备用机上安装设置 **FTP SERVER**，可通过安装 **Internet 服务管理器 (IIS)**，设置 **FTP 服务器**。在本系统中通过 **SERV-U** 设置 **FTP 服务器**，在备用机上设置备份用户及访问口令，设置用户访问目录和权限，存放备份的目录设置为 **D:\backup\sjbf**。

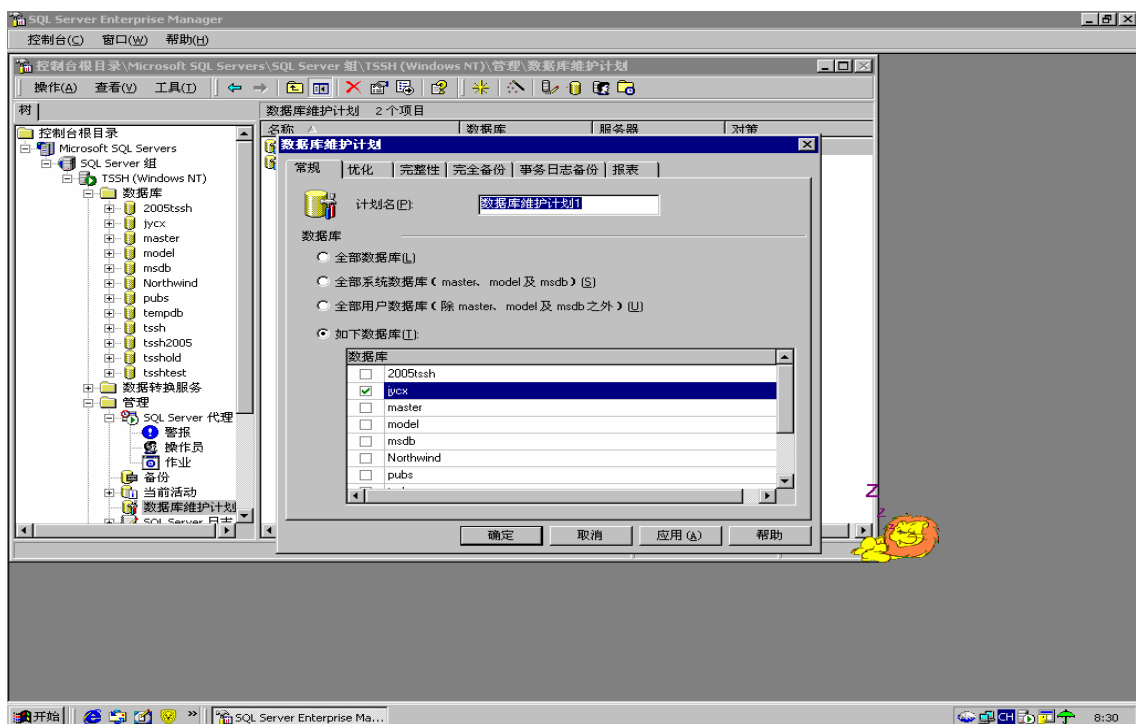


图 4-11 自动备份设置窗口

④利用 FTP 将保存在本地服务器上备份的文件定期上传至设置好的 FTP Server 相应目录下。在目录树中的管理中打开 SQL Server 代理, 右键单击作业选择

新作业，进行新作业属性设置，如图 4-12 所示：

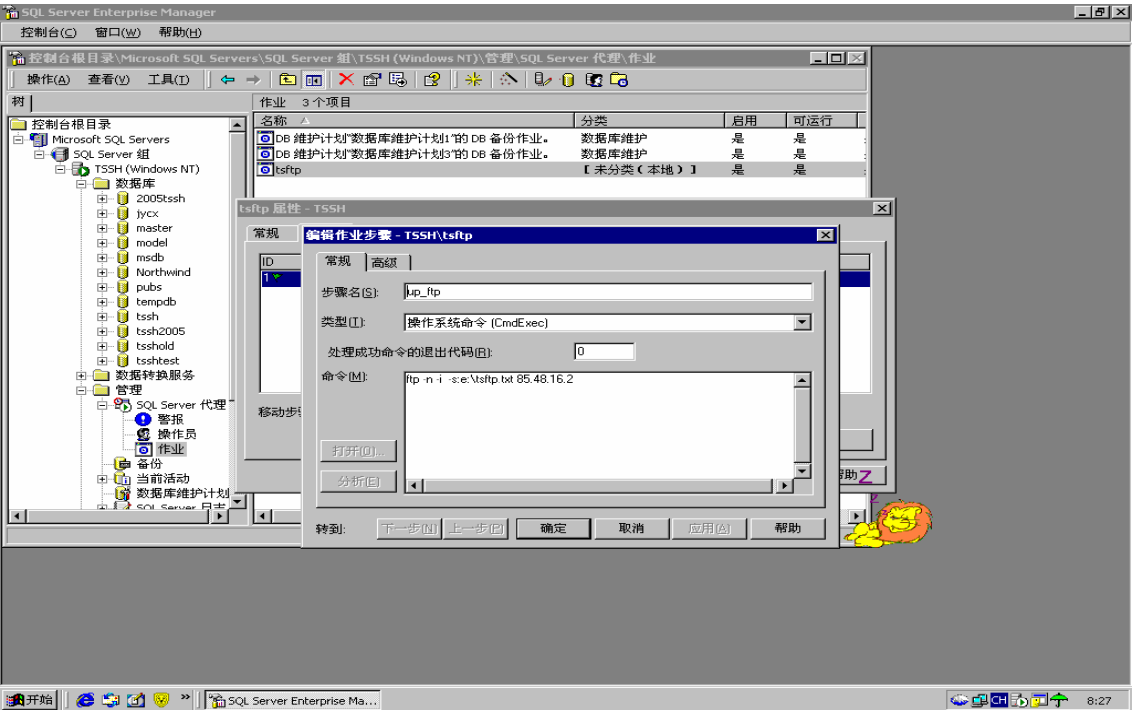


图 4-12 自动传输作业设置窗口

第一步、设置作业名字：在常规标签上名称处输入作业名字，例如 upftp。

第二步、新作业步骤设置，在步骤标签完成。在步骤处输入名字，在类型处选择操作系统命令 (CmdExec)，在命令处输入利用 FTP 上传数据的脚本 `ftp -n -I -A -s:e:\ftp.txt 76.48.16.2`。其中 76.48.16.2 为备用机 FTP 服务器的地址，FTP 服务器必须设置允许写入。e:\ftp.txt 为主用服务器 e:\目录下的一文本文件。其内容为 FTP 命令，定义 FTP 上传的数据，可根据实际情况修改，内容如下：

```
user anonymous ty
cd \
cd backup\sjbf
lcd e:\sjbf
mput *.* /n
bye
```

第三步、调度作业：在调度标签下完成。新建调度，命名，根据需要进行调度设置。

在进行数据库文件备份时，要注意在不同磁盘上对数据库文件及日志文件至少做两个或更多的拷贝。同时要制定好备份策略，经常和定期的作完全备份及增量备份，一旦发生故障，可以方便及时地恢复。

根据不同异常情况，通过备份进行不同程度的恢复，数据库的恢复是数据库



备份的逆过程，只要做好了备份，数据恢复也就没有问题。

#### 4.6.2 基于应用系统的安全技术

基于 C/S 模式下的应用系统安全管理向来是一个众人关注的焦点。而其通常的做法是，直接为每一个系统管理人员或操作人员建立数据库帐号，并为其授权。然而，我们认为这种安全体系有着潜在的问题，尤其是在一个复杂系统中，由于存在着大量的数据库实体及拥有不同操作权限的用户，而每个用户对数据库实体的操作可以是增、删、改、查的任意组合，因此即使使用角色或工作组的方式为其授权，也会显得相当复杂，甚至存在着潜在的安全漏洞。而一旦用户绕过应用逻辑，直接利用自己的帐号操作数据库，那么这些潜在的漏洞将进一步暴露出来，直接威胁系统的安全。

针对这些问题，我们在该系统中所采取的解决方案是<sup>[41,42]</sup>：每个数据库应用只建立一个真正的数据库帐号，它具有对系统应用所涉及的所有数据库实体进行操作的全部权限。与此同时，为每一位系统操作人员分别创建了一个“应用系统帐号”，实际上只是数据库中创建的名为 **USERS** 用户表里的一条记录。

这种安全体系使得应用系统成为数据库的直接用户，而应用系统的所有操作人员（包括系统管理员）则是数据库的间接用户；换言之，应用系统除了完成其应用逻辑之外，还将系统用户和数据库彻底隔离开来，成为数据库的一道坚固的“防火墙”。

然而，正是由于应用系统接管了原来数据库管理系统的一部分工作，因此也增加了一部分工作量。应用程序在与数据库连接中直接在程序中写入数据库用户名和口令，这样虽然方便简单，但是却存在着很大的安全隐患以及管理维护的问题。如果数据库用户口令泄漏或者需要定期修改时，就必须直接修改客户端的源程序，重新编译后才能使用，显然这增加了维护的工作量和难度。为此，必须要把数据库用户口令的使用管理分离出来，做成单独的模块，在系统中采用具有独立性和模块化特点的动态链接库 **DLL** 解决了这个问题<sup>[44]</sup>。应用系统通过设置的只授予默认 **CONNECT** 角色的公用账号 **dbpub** 连入数据库从存有加密用户口令的用户表中取出口令密文，再调用相应动态链接库里的函数解密成明文，然后以客户端需要登入的数据库用户及获得的口令明文登入数据库系统。通过这种方法，如果要改动数据库用户的口令，只需用加密程序把变更的口令生成新的密文覆盖旧的密文即可，还可以方便地更改加密方法，只需重新编制动态链接库覆盖客户端的动态链接库，而客户端的主程序则可以保持不变。从而减轻维护的工作量，加强数据库用户口令的安全。

由于在这种安全体系中，应用系统成为隔离用户和数据库的“防火墙”，其

本身就必须具备相当的安全特性。尤其是用户授权管理机制，其严密性将直接影响整个系统的安全。基于此，我们在业务逻辑层中根据用户的级别和业务范围划分权限，每个用户只能使用自己权限范围内的数据。我们从功能出发将整个系统细分为若干个可分配的最小权限单元，这些权限具体表现在对数据库中所涉及的表、视图的数据操纵（DML:插入、修改、删除、查询等）的划分上和存储过程的执行上。然后再运用角色或工作组的概念，结合各种系统使用人员的工作性质，为系统创建了基本等级：系统管理员、业务操作员，并为每个等级相应地赋予了不同的权限，以此来简化权限管理工作。考虑到系统维护将由信息中心的人员进行，而业务操作由业务人员进行，为了更安全，系统设置为有系统管理员的用户不能有业务权限，有业务权限的不能有系统维护的权限。此外，为了增加系统安全管理的灵活性，授权管理模块还可以对属于某一等级用户的权限作进一步定制，达到所有权限均可任意组合的效果，设置了用户组，可以设置用户组的权限，从而更好地管理用户的权限。根据用户的权限显示用户菜单，只能使用自己拥有权限的操作。

同时，为了进一步提高系统管理员的工作效率，系统为系统权限、用户等级及每种等级所对应的默认权限组合都建立了数据字典，以便在不同的应用环境下，管理员都能方便地增加等级，或改变某种等级的默认权限。并且不允许同一用户在不同的客户端登录。此外，为了能暂时封锁某一帐号的使用，安全系统还提供了帐号冻结及解冻的功能。通过这种方式，在统一管理之下，又具有相当的灵活性，有助于系统管理员更为方便、更为严密地控制整个系统的安全。

#### 4.6.3 基于网络的安全技术

(1)网络边界安全。与 Internet 网物理隔离，在与电信网络接口处设置安全具有加密功能的中网智能防火墙 Lx320。不提供远程 Modem 接入，不设置拨号访问服务器。

(2)网络内部安全控制和防范。每台服务器及客户端均安全全国税务系统专用的自动及时升级的瑞星网络版杀毒软件，安装了入侵检测系统、漏洞扫描系统，全面监控检测系统及网络的活动，确定和堵塞安全及保密的漏洞。

### 4.7 本章小结

本章说明了系统是一个基于网络的、基于 C/S 结构的管理信息系统，对系统的技术方案、数据库设计、信息编码技术、数据库访问技术、系统性能优化以及安全技术等在系统中的具体实现进行了详细的阐述。系统以 Windows2000 Server

为网络操作系统,以 MS SQL Server2000 为后台数据库系统,采用 Powerbuilder9.0 作为客户端应用程序开发工具,论文对它们各自的特点进行了简单介绍;系统的数据库设计权衡性能与存储空间进行规范化和反规范化,综合考虑数据库的完整性,利用 IDENTITY 属性实现了证明单流水号功能;比较 SQL Server 2000 数据库的不同访问方式选择了 SQLOLEDB 方式,对它的定义及原理进行了探讨,研究了 MDAC 组件的构成,实现了系统对数据库的访问;系统性能优化技术主要涉及存储过程及视图使用,以及 SQL 语句的优化;在安全设计中,全面考虑基于数据库的安全、基于应用系统的安全和基于网络的安全,构建了比较完整的安全保护机制。

## 第五章 系统应用

教育储蓄个人所得税管理系统采用 installshield 软件进行打包成 exe 文件, 安装非常简单方便。教育储蓄个人所得税管理系统的界面是通过 Powerbuilder 9.0 来设计创建的, 界面力求简单、可靠、友好。使用简单明了, 设计风格一致, 下面介绍系统的用户界面及测试应用。

### 5.1 系统的用户界面

随着各种各样的软件工具的面市, 软件的用户界面作为软件接口起着越来越重要的作用。用户界面设计的好坏, 从软件开发的角度来看, 直接影响到软件的竞争力。与其他在功能、性能上类似的软件相比, 用户会毫不犹豫地选择具有友好界面的软件; 从用户的角度来看, 直接影响到能否充分的利用信息。

#### 1、可使用性

用户界面的可使用性是用户界面设计最重要也是最基本的目标, 包括以下方面: 使用的简单性、用户界面中所使用术语的标准化和一致性、用户 HELP 帮助功能、快速的系统响应和低的系统成本、用户界面具有容错能力。

#### 2、灵活性

考虑到用户的特点、能力、知识水平, 用户界面能满足不同用户的要求; 用户可以根据需要制定和修改界面的方式; 系统能按照用户的需求, 提供不同详细程度的系统响应信息; 具有标准的界面。

#### 3、复杂性和可靠性

复杂性是指用户界面规模和组织的复杂程度, 在完成预定义的功能的前提下, 应该使用户界面起简单越好; 而可靠性是指系统无故障使用的时间间隔。用户界面应保证用户正确、可靠的使用系统, 保证有关程序和数据的安全性。

教育储蓄个人所得税管理系统的界面是通过 Powerbuilder 9.0 来设计创建的, 界面力求简单、可靠、友好, 方便用户使用。其界面设计有以下特点。

1、界面设计简洁明了, 以实用为标准。

2、系统各个模块界面统一, 风格一致。

3、系统所有提示、按钮、菜单一律使用标准汉语词汇及标准税务词汇。尽量简洁明了, 支持使用键盘和鼠标都可以操作。系统按钮尽量使用两个汉字的按钮, 并且汉字之间不能留有空格。

4、所有操作全面支持鼠标和键盘, 适应不同层次人员使用需求。

5、屏幕使用标准屏幕大小, 800\*600 分辨率或以上。

6、字体全部使用系统字体。

7、合理使用颜色，同时保证系统的统一性。以下列举系统中几个用户界面加以说明。

(一)、登录，输入用户名和口令按确定登入系统，不同权限用户操作界面不同，不同税务机关操作的数据也不相同。



图 5-1 登录窗口

(二)、主菜单

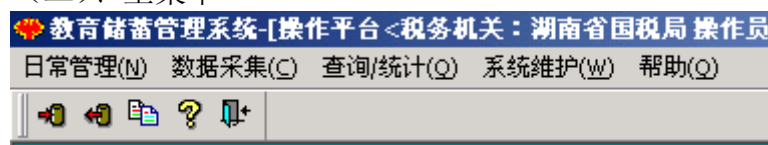


图 5-2 主菜单窗口

(三)、数据采集，导入从银行申报的开户信息，流水帐信息，证明单信息三类数据，并提供查看采集情况的功能。点击“导入”按钮弹出导入数据文件窗口，选择要导入的数据文件。还可以进行申报情况的查询。

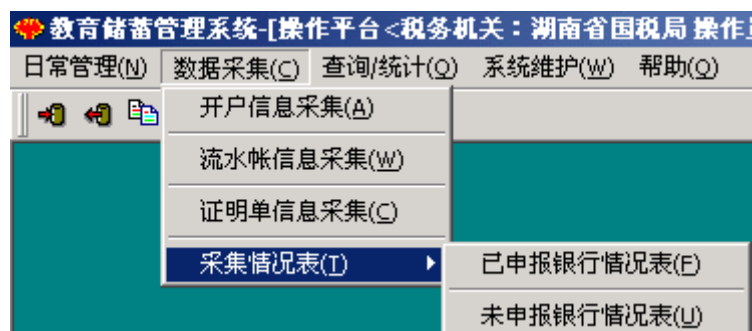


图 5-3 数据采集窗口

- (四)、日常管理
- 1、数据比对，点击数据比对菜单时弹出如下窗口，并将未申报的银行显示出来，点“比对”按钮将进行数据比对。可进行比对情况的查询。



图 5-4 数据比对窗口

- 2、证明单管理
- 证明单管理菜单：市级用户对应县（区）局用户，县（区）局用户对应学校。可进行证明单入库、证明单发放和退回的操作，证明单一本 25 份，输入起始号码及本数系统自动产生终止号码，点后“入库”按钮。入库后证明单即可以下发，市局下发给县（区）局，县（区）局下发给学校。未使用的证明单可以退回。

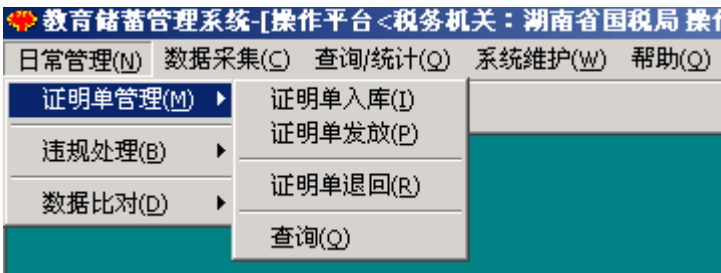


图 5-5 证明单管理窗口

- 3、违规处理
- 实现银行罚款、学校罚款、查补利息所得税、异常账户处理等功能。税源管理科处理方式分为转为正常户、转为非教育储蓄帐户、移交稽查局处理 3 种。稽查部门处理方式为转为正常户、转为非教育储蓄帐户 2 种。

- (五)、查询、统计
- 包括 4 类共 10 个统计表。以教育储蓄余额明细表为例。



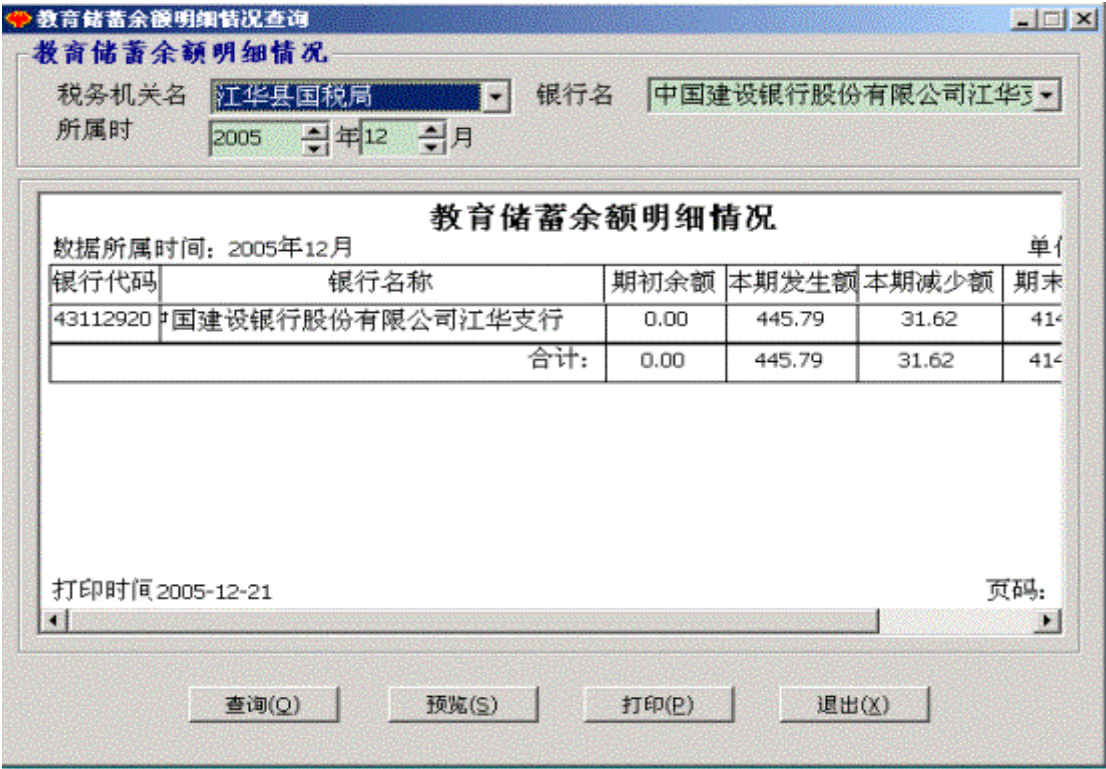


图 5-6 查询窗口

(六)、系统维护

实现各代码初始化、用户及用户组管理、权限分配、数据备份及恢复、账号锁定及解锁等功能。



图 5-7 银行代码初始化窗口



图 5-8 权限分配窗口

## 5.2 系统测试及应用

系统测试是为了发现错误而执行程序的过程，成功的测试是发现了至今尚未发现的错误的测试。系统测试是保证系统质量和可靠性的关键步骤，是对系统开发过程中的系统分析、系统设计和实施的最后复查。在进行信息系统测试时应遵循的原则：

- 1、应尽早并不断进行测试。
- 2、测试工作应该避免由原开发软件的人或小组承担。
- 3、设计测试方案的时候，不仅要确定输入数据，而且要根据系统功能确定预期输出结果。
- 4、在设计测试用例时，不仅要设计有效合理的输入条件，也要包含不合理、失效的输入条件。
- 5、在测试程序中，不仅要检验程序是否做了该做的事，还要检验程序是否做了不该做的事。
- 6、严格按照测试计划来进行，避免测试的随意性。
- 7、测试用例都是精心设计出来的，可以为重新测试或追加测试提供方便。

软件测试方法分人工测试和机器测试。人工测试指的是采用人工方式进行测



试，目的是通过以程序静态结构的检查，找出编译时不能发现的错误。又称为代码审查。机器测试是用设计好的测试用例作用于被测程序，比较测试结果和预期结果是否一致，如果不一致，就说明可能存在错误。机器测试只能发现错误的症状，但无法对问题进行定位。机器测试又分为黑盒测试和白盒测试。

在教育储蓄个人所得税管理系统的测试过程中经过专业人员的代码审查后，根据各个模块接口、数据结构、重要的执行路径、出错处理、边界条件等特点，精心设计了测试用例，同时还利用了 SQL 语句中的随机数函数来生成不同的身份证号、银行帐号、储户姓名等关键信息，按照 10% 的违规比例再生成相应的违规信息，总计生成数据约 4 万条左右，对系统进行了各种测试，解决测试中出现的各种错误。在 P4 3.0G+512MB 内存的普通台式电脑上运行比对算法，平均每次耗时 12 秒，比对的违规信息符合业务需求，各功能模块均完成其功能。通过对后台输出的调试信息比较，发现主要时间开销在于磁盘 I/O 开销，如果能换用 SCSI 接口的服务器，性能应该会有所提高。

经过单元测试、组装测试及确认测试的有效性测试后，在永州市全市国税系统范围进行了两个月的验收测试，按用户要求的真实环境进行安装，按正常的业务流程处理业务，测试以用户为主，由软件开发人员和业务科室的人员共同参与，由用户参与设计一切可能的测试用例，测试时采用实际数据及暂时没能遇到的预计可能发生的数据，同时对系统的安装备份恢复、安全性、强度、性能、可靠性进行了测试。针对测试时发现的问题进行了调试，找出原因和具体的位置，进行改正。并对用户提出的新的需求进行了完善。

通过测试和调试的教育管理系统于 2006 年 1 月在永州市全市范围内试运行，服务器采用浪潮 7410 服务器，处理器 Intel®Pentium®III Xeon™800MHz；内存 1G；，带 RAID 卡，36G\*5 SCSI 硬盘存储空间，使用 RAID5 确保系统在节省开销的同时获得良好可用性和性能，连接 UPS 电源，双网卡 10/100Mbps 自适应快速以太网卡。双 4M SDH 电信网络。税务版服务器安装于市局，数据集中于市局，由市局进行数据初始化。税务版客户端安装于市局相关业务科室及各县区纳税大厅相应窗口。

由于系统用户界面友好，数据库连接不需特殊的设置，用户权限明晰，不用的用户拥有不同的操作菜单，操作简便，培训到位，系统帮助文件详细，推广顺利。试运行三个月，查出现有的教育储蓄有三分之一以上属化整为零、超限额存款、多银行存款、多地域存款、无证明单等违规存款，共查补利息个人所得税 300 余万元，给违规的储户、银行及学校很大的震慑作用，满足了日常征管的需要，使教育储蓄利息个人所得税的征管步入规范化，达到了预期的目标，用户反映良好。

### 5.3 本章小结

本章用实例说明了系统用户界面的设计原则和设计特点，并对系统的测试及实际应用效果进行了简单的说明。

## 第六章 结论及展望

基于 C/S 结构的教育储蓄个人所得税管理系统的设计开发,体现了基于 C/S 结构的应用模式,采用了 OLE DB 数据库访问等技术,这些应用模式和技术具有强大的功能和普遍的适用性,因此,通过本系统的开发不仅完成了一个具有实用价值并且已经投入试运行的应用系统,而且利用该系统的模式和技术可以较方便地完成类似的其它资源、数据的管理,为今后的工作打下一个良好的基础。在该项目的设计与开发中,主要完成了以下工作:

- 1、较好地实现了用户、开发人员、以及其他涉及项目人员之间的信息交流,在对系统业务规则充分理解和描述的基础上,对系统的整体结构(包括网络结构和功能结构)进行了规划。

- 2、权衡性能与存储空间进行规范化和反规范化,综合考虑数据库的完整性完成系统的数据库设计,利用 IDENTITY 属性实现了证明单流水号功能;

- 3、比较 SQL Server 2000 数据库的不同访问方式选择了 SQLOLEDB 方式,对它的定义及原理进行了探讨,研究了 MDAC 组件的构成,实现了系统对数据库的访问;

- 4、系统性能优化技术主要涉及存储过程及视图使用,以及 SQL 语句的优化;

- 5、在安全设计中,全面考虑基于数据库的安全、基于应用系统的安全和基于网络的安全,构建了比较完整的安全保护机制。

目前,教育储蓄个人所得税管理系统已经投入使用,在湖南省永州市国税系统运行已经有三个多月时间,取得了明显的成效。

- 1、提升了教育储蓄管理信息化水平。实现了教育储蓄管理的信息化管理,使教育储蓄税收管理由人工粗放型管理逐步转向全过程计算机控制型的管理,弥补了 CTAIS 和湖南征管系统未将教育储蓄纳入系统管理的不足,使教育储蓄管理在信息化的进程中又迈进了一大步。

- 2、加强了储蓄存款利息所得个人所得税征收管理,规范了教育储蓄利息所得免征利息税管理,解决了各金融机构之间信息不能共享而无法真正进行管理的的问题,确保了《储蓄存款利息所得征收个人所得税的实施办法》和《教育储蓄管理办法》等教育存款利息所得免征个人所得税政策的正确贯彻执行。

- 3、规范了执法行为,树立了国税形象。实现了用现代化的科技手段控制税收人员的执法行为,最大程度地减少了人为因素的影响,减少了执法的弹性,推动了税务部门依法行政。

4、税源监控力度加大,管理效率得到提高。教育储蓄个人所得税管理系统的推行,实现了计算机控管,使管理效率明显提高。系统将各金融机构的储户开户、存取款、证明单等信息导入管理系统,定期进行比对,将可能违规的银行、储户和学校信息找出来进行监控分析,并进行及时处理。有效扼止了各金融机构储蓄网点将教育储蓄作为揽储的一种重要手段,违规为储户开办教育储蓄的行为。也有效控制了学校开具非义务教育证明的失控现象。有效监控了银行、储户和学校的教育储蓄免征个人所得税的行为,强化了扣缴义务人履行代扣扣缴义务。

5、有利于增加税收收入,减少税收流失。通过教育储蓄个人所得税管理系统,加强的税收征管,推进了依法治税,堵塞了教育储蓄免税管理漏洞,避免了违规教育储蓄存款利息个人所得税的流失,增加了税收收入。从而也真正地达到国家教对育储蓄存款利息所得免征个人所得税这项优惠政策的目的,确保了合法储户的利益,确保了税法的严肃性。

尽管课题的研究已经取得了初步的成果,但由于新应用的不断出现和计算机技术的快速发展,在本文研究的基础上,下一步要重点开展的工作是:

1、进一步研究系统在全省范围的使用,如果在全省使用可以进一步比对所有范围的数据,可以查处在不同的地市违规进行教育储蓄的储户、银行和学校。这将要进一步对系统进行性能优化,还要考虑如何避免存储空间和不足。

2、进一步研究与中国税收信息征管系统(CTAIS)的数据交换,达到共有数据的共享和同步。现在数据交换技术很多,其技术也逐步走向了成熟。如 XML 是一种通用的结构化数据表示方法,它允许应用程序存储和传输那些能被其它应用程序理解的数据,并将数据的格式和内容与处理方法分离开来。并克服了传统的数据交换系统的缺点。

3、进一步研究系统基于 B/S 模式的转化和改进。从今后的发展趋势看,Internet/Web 是集成各种信息技术和成果的合适框架,而 Browser/Server 体系结构紧密结合了 Internet/Intranet 技术,是未来的发展方向。Browser/Server 模式以其良好的可扩展性、超越地理位置限制等优势,成为管理信息系统发展的必由之路。本系统基于 C/S 模式的结构也必然向基于 B/S 模式的结构转化。

4、进一步研究系统的安全性问题。一个分布式系统的性能,不仅仅取决于其本身系统设计的好坏,如对象的分布是否合理、网络通信机制是否高效以及系统的性能,还取决于该系统的安全性。

## 参考文献

- [1] 国家税务总局. 税务管理信息系统一体化建设总体方案. 国家税务总局, 2001
- [2] 国家税务总局, 中国人民银行等. 教育存款利息所得免征个人所得税实施办法, 2005
- [3] 孙开盛, 韩建民. 基于C/S模式的分布式数据库管理系统的研究与实现. 微计算机应用, 1998(9)
- [4] Darleen Sadoski. GTE Client/Server Software Architectures--An Overview, [http://www.sei.cmu.edu/str/descriptions/clientserver\\_body.html](http://www.sei.cmu.edu/str/descriptions/clientserver_body.html), 1997
- [5] 刘独玉, 刘彬. 基于MS SQL Server的存储过程的研究与应用. 四川轻化工学院学报, 2001, 6(2)
- [6] 联想神州数码公司. 中国税收征管信息系统操作手册. 北京: 国家税务总局, 2002, 10-15
- [7] 陈康, 向重伦, 蔡学望. 税务管理信息系统研究: [硕士学位论文]. 成都: 西南财经大学, 2000
- [8] Paul Hrabal. DisasterRecovery White Paper. <http://www.usdatatrust.com>, 2004.1
- [9] 梁欣. 证券业预约呼出系统的设计与开发: [硕士学位论文]. 长沙: 中南大学, 2004
- [10] 胡义勇. 基于XML的电子缴税系统的设计与实现: [硕士学位论文]. 长沙: 中南大学, 2005
- [11] 余欢. 网络环境下容灾备份技术在国税系统中的应用研究: [硕士学位论文]. 长沙: 中南大学, 2005
- [12] 张聚礼, 余冬梅等. 基于C/S的软件体系结构研究. 计算机工程与应用, 2003(8), 58~60
- [13] 勾建新. 数据库的性能优化技术. 湖北汽车工业学院学报, 2003, 17(3), 30~34
- [14] Adler, R. M. Distributed Coordination Models for Client/Sever Computing. Computer 28, 4 (April 1995): 14-22.
- [15] 李劲. Windows2000 Server组网与安全配置手册. 北京: 中国青年出版社, 2000(7)
- [16] 张权, 王家顺等. 企业信息编码技术的研究. 计算机工程. 1998, 28(7)
- [17] 黄新亚, 米央编. 信息编码技术及其应用大全. 北京: 电子工业出版社, 1994
- [18] 黎连业等. 计算机管理系统设计与实现. 北京: 学苑出版社, 1993
- [19] Schussel, George. Client/Server Past, Present, and Future [online]. Available WWW <URL: <http://www.dciexpo.com/geos/>> (1995).

- [20] Gallagher, J. & Ramanathan, S. "Choosing a Client/Server Architecture. A Comparison of Two-Tier and Three-Tier Systems." *Information Systems Management Magazine* 13, 2 (Spring 1996): 7-13.
- [21] 曹维远, 叶文川. C/S模式和B/S模式交叉并用的MIS系统平台. *微计算机应用*. 1999,20(2):65~68
- [22] 郭宝利. *Power Builder 函数及应用*. 北京: 北京达通兴电脑科技公司, 2004
- [23] 张月琳. 教育管理信息系统开发环境的选择. *管理信息系统*, 1997(10) : 19~21
- [24] 林郁. 数据库的备份及恢复技术. *计算机应用*, 1994 (10), 61~62
- [25] 丁正刚, 张全秋. SQL Server2000数据库服务器与PowerBuilder9.0连接研究. *福建电脑*, 2004(3), 70~71
- [26] 杜磊, 秦耕等. 基于RAS的C/S应用原理与实现. *计算机工程*, 2000, 26 (11)
- [27] Dorian Coughias, E L Heiberger, Karsten Koop. *The Backup Book: Disaster Recovery from Desktop to Data Center*. New York: the network frontiers, 2003
- [28] Robert Horst. IP Storage and the CPU Consumption Myth: Network computing and Applications. *IEEE international symposium*, 2001: 194-200
- [29] 邓健青, 石岗. 应用程序中数据库的备份与恢复的实现. *计算机应用研究*, 2004(5), 143~144
- [30] (美)Edward Whalen等著. *SQL server 2000性能调整技术指南*(武欣, 何畅, 罗云峰等译). 北京: 机械工业出版社, 2002
- [31] 巩建国, 裴红等. 数据库服务器的安全性. *山东建建筑筑工程学院学报*, 2004, 19(2), 80~83
- [32] 郑谦益. Sybase SQL server 性能优化技术及应用研究. *微机发展*, 2003, 13 (1)
- [33] 金淼. 浅议银行数据大集中后的数据安全. *网络安全技术与应用*, 2004(5), 48~49
- [34] MITCHELL F.H. *CIM System - an introduction to computer integrated manufacturing*[M].prentice - Hall International, 1991
- [35] 林志斌. 数据库安全性若干问题的探讨. *微型机与应用*, 1998(3)
- [36] 陈从锦, 杨昱. 数据库的安全防护概述. *中山大学学报*, 2003, 23(3): 208~211
- [37] 潘群华, 景宁. 数据复制及备份技术研究[硕士学位论文]. 长沙: 国防科学技术大学. 2002
- [38] [美]John papa Malthew shepker等. *SQL Server7编程技术内幕* (M). 机械工业出版社, 2000(1)

- [39] 袁鹏飞. SQL Server 2000数据库系统管理与应用开发[M]. 人民邮电出版社, 1999
- [41] H.M.Gladney. Data Replicates in Distributed Information Services. ACM Transaction on Database Systems, 1989, 14(1): 75-97
- [42] William Premerlani, Michael Blaha. An approach for reverse engineering of relational database. Communications ACM , 1994, 37(5): 42-49
- [43] Microsoft Corporation. Microsoft SQL Server 2000 data operation and replication. America: Microsoft Press, 2001. 3-51
- [41] 王晋东, 张明清, 韩继红. 信息系统安全技术策略研究. 计算机应用研究, 2001, 15(5)
- [42] 朱虹. DBMS的安全管理. 计算机工程与应用, 2000 , 36(1)
- [43] 刘波. 管理信息系统中数据库安全实现方法. 计算机应用, 2000, 20(10)
- [44] 罗戈夕. 数据加密解密处理的设计及实现. 计算机应用, 2004(7):64~65
- [45] <http://www.cnask.com>; <http://msdn2.microsoft.com>; <http://searchdatabase.techtarget.com.cn>等

## 致 谢

首先，我深深地感谢我的导师桂卫华教授，桂教授知识渊博、为人和善、严于律己、宽以待人，在许多方面都是我学习的榜样。能在这样的导师指导下学习、工作，我感到非常的幸运。正是他的关心与指导，使我能够较好地完成各项工作。在此论文完成之际，谨向桂教授致以最诚挚的敬意和衷心地祝福。

在项目开发及论文的完成过程中，湖南省国税局张社安副局长给了我大量的指导和帮助，在此深表谢意。

同时，在项目开发过程中，开发组的其它同志给了我许多帮助，我们科室和许多业务科室的领导和同事给予了大力支持和帮助，谨以此表示深深的感谢。同时还感谢同学及亲人给予我的大力帮助。

我还要感谢省国税局和各级领导，感谢他们为我们创造了这次学习的机会和良好的学习环境！感谢我所在单位的领导和同事们，几年来是他们毫无怨言地自始至终地支持着我！

最后，向所有支持我、关心我和帮助我的家人、老师、同学和朋友表示衷心的感谢！衷心感谢关心和帮助我的每一个人！

2006 年 4 月于中南大学



## 硕士研究生期间发表论文

- [1]吴芙蓉, 桂卫华.数据挖掘及其在税务税额核定中的应用.计算机技术与自动化,2004, 23 (3), 269~271