

The background features a decorative graphic consisting of three overlapping circles in shades of blue, arranged along a diagonal line. The circles are positioned in the upper right, middle right, and lower right areas of the page. The text is centered horizontally and positioned between the upper and middle circles.

分布式数据库课程设计和实现

——面向连锁超市管理系统分布式数据库设计和实现

目录

1. 连锁超市管理系统概述	1
1.1 背景介绍	1
1.2 可行性分析.....	1
1.3 系统目标和优异之处	2
1.4 小组成员和任务分配.....	3
2. 需求分析	4
2.1 用户需求概述	4
2.2 业务需求分析.....	5
2.3 功效分析	6
3. 系统总体设计	8
3.1 绘制用例图设计系统功效	8
3.2 绘制系统步骤图	9
3.3 系统开发环境.....	10
3.4 系统运行环境	11
3.5 系统运行演示.....	11
3.6 系统类库设计	14
4. 分布式数据库分析和设计.....	16
4.1 数据库分析	16
4.2 数据库概念设计	16

4.3 数据库逻辑结构设计	20
4.4 分片和位置分配设计	24
4.4.1 站点通信模型	24
4.4.2 数据表分片和位置分配设计	25
5. 基于 SQLSMS 具体设计（实现）	27
5.1 公共类设计	27
5.1.1 文件操作类 CFileOperate	27
5.1.2 时间格式转换类 CMyTime	29
5.1.3 数据操作类 CADConn	29
5.1.4 数据库表操作类 CtablePartInfo	37
5.2 界面窗口设计	41
5.2.1 系统开启登录开发	41
5.2.2 主界面对话框设计	43
5.2.3 通用报表对话框设计开发	44
5.2.4 打印对话框设计开发	46
5.3 基础资料模块设计	48
5.3.1 基础资料管理功效开发	48
5.3.2 价格自动生成功效开发	52
5.3.3 报表功效实现	54
5.4 进货/销售模块设计	55
5.4.1 进货/销售登记设计	55
5.4.2 进货设计	59

5.4.3 月度统计设计	61
5.5 库存模块设计	62
5.5.1 库存查询功效开发	62
5.5.2 库存报警功效开发	63
6. 开发技巧和难点分析	64
6.1OLE 技术使用	64
6.2MFC GRID CONTROL 控件使用	65

1. 连锁超市管理系统概述

1.1 背景介绍

伴随市场经济发展和人民生活水平提升，原来单一、小规模超市已无法满足人民对购物环境要求，大规模、物品丰富超市正在蓬勃发展。超市销售数据规模日益庞大，商品数目标快速增加，采取以往手工管理已直接或间接地降低了工作效率，最终影响超市日常运转。

另外超市发张壮大，尤其是连锁超市（如家乐福、美廉美）扩张，使得她们含有地域上分散而管理上又相对集中特点，往往既要有各门店局部控制和分散管理，同时也要有整个组织全局控制和高层次协同管理。

所以把这些门店和中心经过网络连接起来，设计开发一款基于分布式数据库连锁超市管理系统 SMS（Supermarket management system）势在必行。

1.2 可行性分析

（1）经济可行性

使用连锁超市管理系统对超市连锁店进行信息化管理将直接提升企业管理部门工作效率。经过网络远程提交汇总各门店信息，节省了很多时间和金钱。其次，用信息化管理统计大量数据，节省了很多人力和财力，为管理者提供愈加好决议支持。

（2）技术可行性

网络应用基础设施完善，因为信息技术发展，中国计算机网络飞速发展，前后建成了中国公众多媒体通信网、China Net、中国教育和科研计算机网络等组成了中国 Internet 主体，网络应用进入企业和一般家庭，这为发展连锁超市网络办公提供了基础设施。

网络安全技术应用，包含加密算法、CA 数字认证、数字署名等，为网络办公系统应用提供安全确保，它实现了网络传输数据安全性、完整性等。

最终，就是网络技术普及和掌握，如网络互联、网络安全技术、网络数据库技术等，使我们有能力开发实现适合自己管理系统。

所以，企业构建跨区域管理系统技术瓶颈问题（网络应用基础设施、网络安全、开发技术）得到了有效处理，企业构建网络管理系统技术上可行。

（3）操作可行性

因为 SMS 操作是基于 C/S 用户端页面操作，简单明了，用户无需学习，通常全部能够很轻易知道怎样操作。而管理员也无需含有专业知识，只需要对部分数据进行输入和平时日常维护就够了。

1.3 系统目标和优异之处

（1）系统目标

为连锁超市提升效率、降低成本；实现连锁超市管理优化，简化工作步骤，节省人力物力，提升工作效率，极大地满足用户需要；对各步骤进行控制分析，实现统一调度。

满足连锁超市基础管理功效，发挥信息系统灵活性，减轻企业管理人员和操作人员工作负担，提升工作效率。

本系统是一项功效比较完善连锁超市管理系统，对连锁店运作过程中后台数据能够随时进行分析，便于企业管理人员经营、决议。全方面表现了现代企业管理理论所提倡工作高效、环境轻松气氛。

(2) 系统优异之处

- 多数处理就地完成。
- 各地计算机由数据通信网络相联络。
- 克服了中心数据库弱点：降低了数据传输代价。
- 提升了系统可靠性，局部系统发生故障，其它部分还可继续工作。
- 各个数据库位置是透明，方便系统扩充。
- 为了协调整个系统事务活动，事务管理性能花费高。

1.4 小组成员和任务分配

2.1 用户需求概述

下面是经过调查研究取得相关连锁超市关键信息数据需求分析结果：

(1) 连锁超市通常由一个中心（企业总部）、多个远程连锁店（门店）组成，而且每个连锁店分布在不一样地域。

(2) 连锁超市各个部门之间、各个分店之间、分店和总部之间需要交换数据，这种数据交换是经过局域网和广域网进行。

(3) 企业总部负责产生并管理该连锁超市整体汇总数据，即各门店明细汇总表数据，如销售汇总表等。

(4) 每一个远程站点（各门店和企业总部）分别有一个数据库系统，各自组成一个独立子系统，能够分别独立进行本部门业务处理。

(5) 总部为了便于对各店进行管理，同时也为了比较各店销售情况，要求门店将全部商品归入对应商品类别，由总部统一管理并提供各门店使用，而且，商品类别信息数据在各门店全部要常常使用。

(6) 相关商品信息、供给商信息、POS 机信息、进货信息和销售信息等经营基础数据全部是各门店单独管理和使用，门店之间互不相关。

(7) 整个连锁超市职员信息由企业总部管理和维护，各门店只能够查询本部门职员信息。

2.2 业务需求分析

经过对超市营业、管理业务步骤调查，得到下面业务需求。

- 能够对商品类别、商品、供给商、POS 机等基础信息进行管理。
- 能够实现采购开单、销售开单、采购退货、销售退货等基础功效。
- 软件可依据商品资料中设置最高库存、最低库存进行库存报警。
- 商品批发价、零售价可自动生成，减轻定价工作量。

- 能够生成各类统计报表，提供详尽营业汇报，实现对商品进、销、存及利润等财务情况了如指掌。
- 全部单据、报表均能够在打印前预览，而且能够导出为 Excel 文件，然后依据实际需要进行尤其编排处理。
- 为了系统安全，每次用户登录全部创建对应日志文件，统计用户全部操作。

另外，对系统性能关键有以下多个方面需求。

系统在设计过程中应充足考虑到可扩充性，要求操作界面美观大方，轻易上手。

2.3 功效分析

依据对系统业务调查和用户需求分析，结累计算机信息管理特点，设计系统实现功效以下。

（1）基础信息管理

基础信息管理包含业种商品类别信息、商品信息、供给商信息、POS 机信息和价格信息维护和管理，实现功效包含：

- 基础信息添加、删除和更新操作。
- 生成各类基础信息报表。
- 打印和导出报表。

（2）进货管理

进货管理关键实现对商品采购入库信息进行管理，具体实现功效以下：

- 进货开单，实现商品进货结算、入库操作。
- 进货退货，退还商家相关商品。
- 生成进货、退货单据、商品报表。
- 打印和导出报表。

（3）销售管理

销售管理关键实现对商品销售出库相关信息进行管理，具体实现功效以下：

- 销售开单，实现商品销售结算、出库操作。
- 销售退货，许可用户退出相关商品。
- 生成销售、销售退货单据、商品报表。
- 打印和导出报表。

（4）库存管理

库存管理关键实现对商品库存相关信息进行管理，具体实现功效以下：

- 库存查询，能够查看全部库存商品相关信息。
- 库存报警，对库存过多或过少商品进行报表统计。
- 库存盘点，能够修改商品库存数量。

（5）帐务管理

帐务管理关键实现对营业员销售商品、业务分成、营业收入等情况进行分类报表统计。

（6）数据管理

数据管理关键实现对数据库数据进行备份、还原及清理等相关工作。

(7) 系统管理

系统管理关键实现登录用户（职员）管理、系统日志、修改登录密码等相关工作。

2. 系统总体设计

3.1 绘制用例图设计系统功效

用例图表示了角色和用例和它们之间关系。它描述了系统、子系统和类一致功效集合，表现为系统和一个或多个外部交互者(角色)消息交互动作序列。也就是角色(用户或外部系统)和系统(要设计系统)为了实现一个目标交互，这个目标描述通常是一个谓词短语，比如签合同相同。

系统设计包含超级管理员、管理员和营业员四种用户角色。超级管理员含有全部操作权限，其用例图图 3-1 所表示。

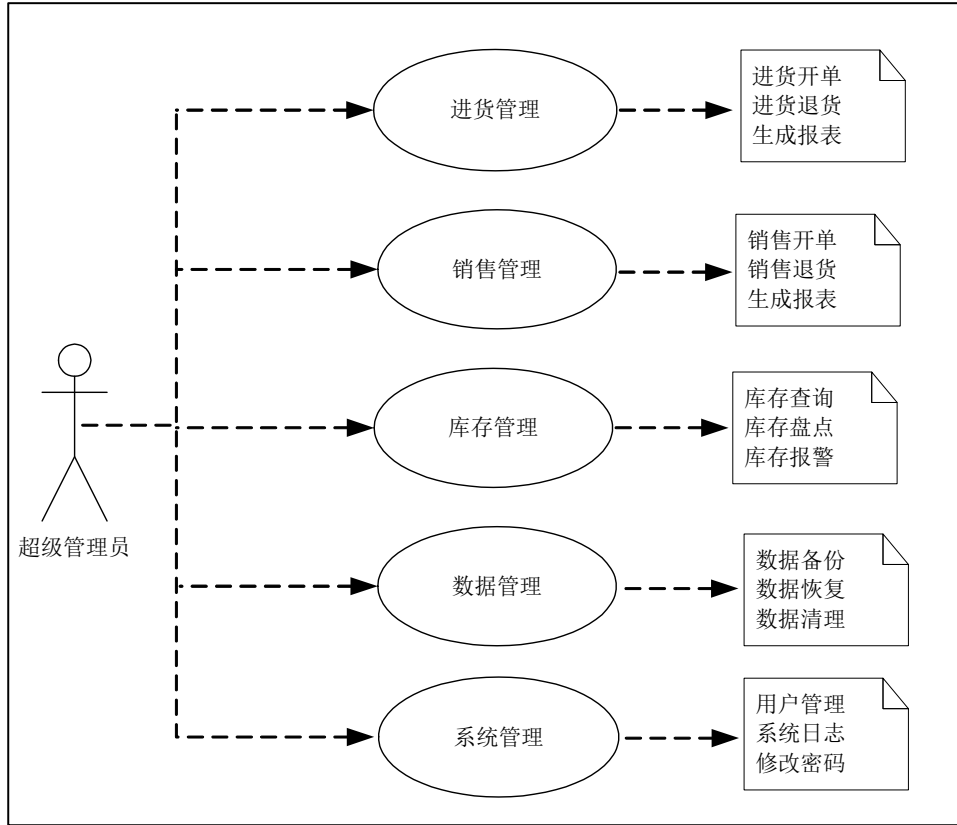


图 3-1 超级管理员角色系统用例图

管理员不含有系统设置模块中用户管理和商品类别信息管理功效，其它功效均含有。而销售人员则只含有销售开单管理功效。

3.2 绘制系统步骤图

结合系统具体设计要求，连锁超市管理系统关键功效步骤图图 3-2 所表示。

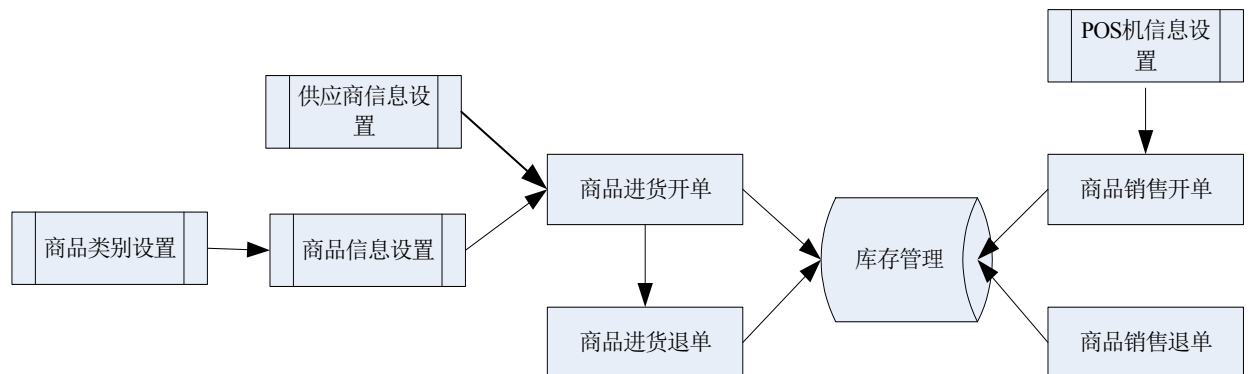


图 3-2 系统关键功效步骤图

系统首先对登录用户身份进行验证,依据用户权限激活相关功效。

超级管理员含有全部操作权限,系统功效结构图图 3-3 所表示。

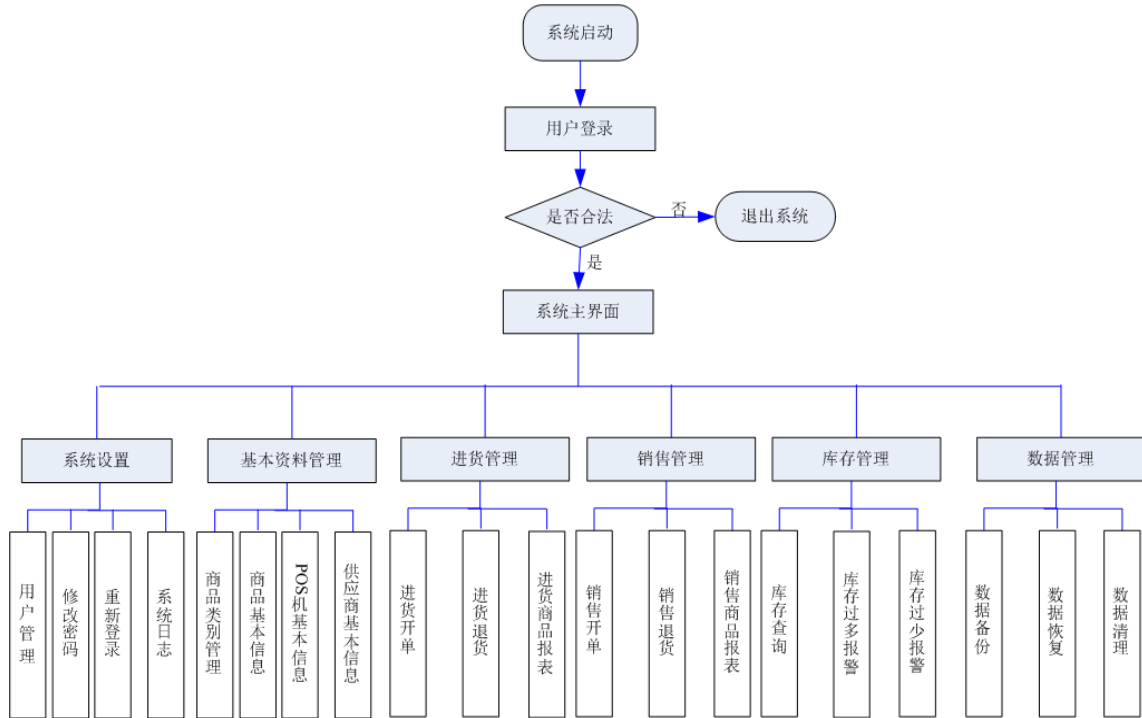


图 3-3 系统功效结构图

上图只是列出了部分关键功效,系统还能够依据不一样需求生成多种统计报表。

3.3 系统开发环境

本系统是在 Windows XP 汉字版操作系统环境下,使用 Microsoft Visual Studio 汉字版用 C++语言开发成功。在开发过程中,使用了 OLE 技术和 ActiveX 控件技术。

后台数据库系统设计采取是 Microsoft SQL Server 和 Access 数据库系统,经过 ADO 数据库开发技术,直接操作数据库文件。

3.4 系统运行环境

系统能够直接在 Win98、Win、WinXP 环境下运行。系统预设超级管理员用户名为“admin”，密码为“admin”。

3.5 系统运行演示

程序开启，首先弹出图 3-4 所表示“系统登录”对话框，只有输入正确用户 ID、登录密码才能进入系统能够。



图 3-4 “系统登录”对话框

假如登录用户为超级管理员或管理员，会进入系统主界面窗口，图 3-5 所表示。

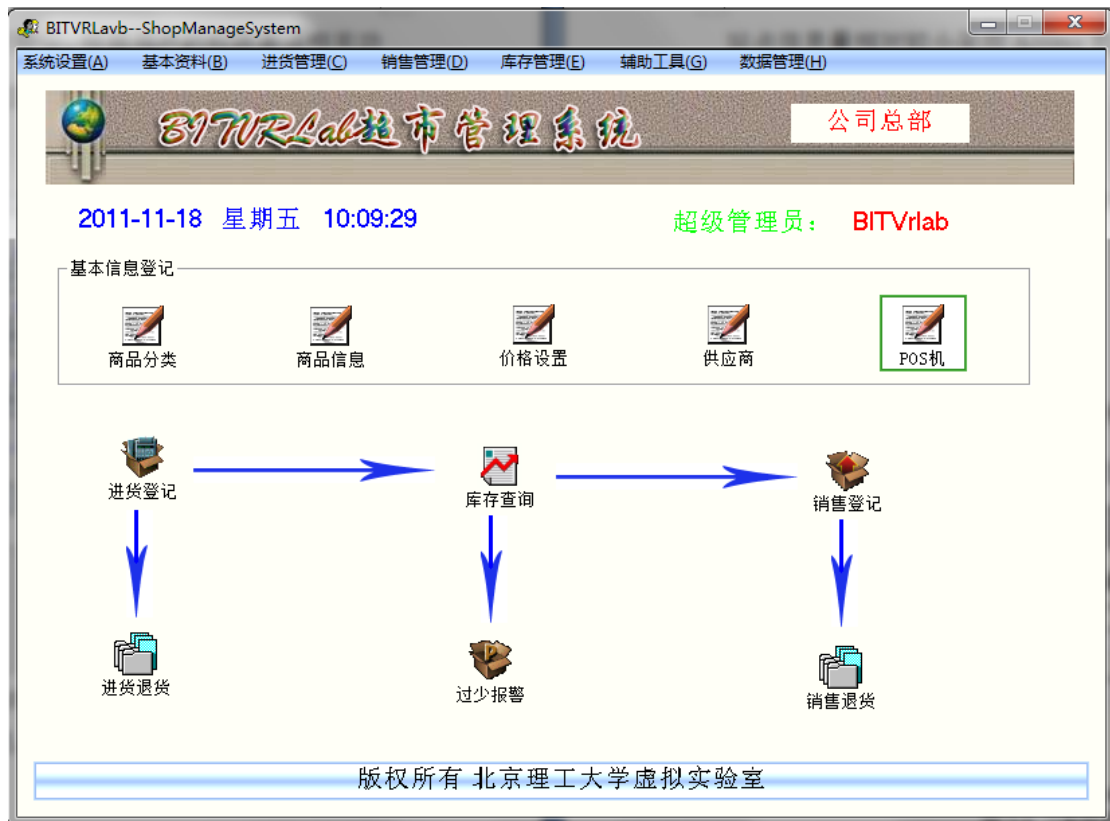




图 3-5 系统主界面窗口

假如登录用户权限为营业员，因为其只含有售货权限，所以系统会直接弹出“销售开单”对话框，图 3-6 所表示。

销售开单


销售开单登记管理

单据编号: 开票日期:

经办人:  分机号: 

备注: 区域:

加载商品

商品检索: 

(输入条形码/编码/助记码/拼音码进行检索)

条形码	名称	库存上限	库存下限	销售价

销售价: 数量: 总金额:

图 3-6 “销售开单”对话框

在进行相关操作前，首先需要添加、设置部分基础资料，如商品分类登记，其操作设置对话框，图 3-7 所表示。

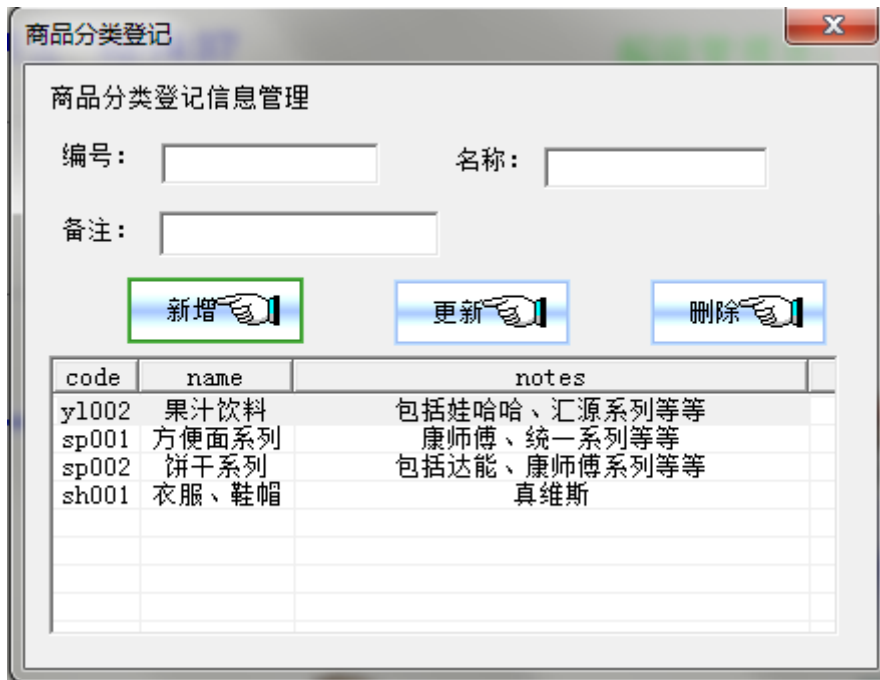


图 3-7 “商品分类登记”对话框

另外，系统还提供了丰富报表功效，图 3-8 所表示商品分类报表，同时对报表提供了导出到 Excel 文件和打印功效。

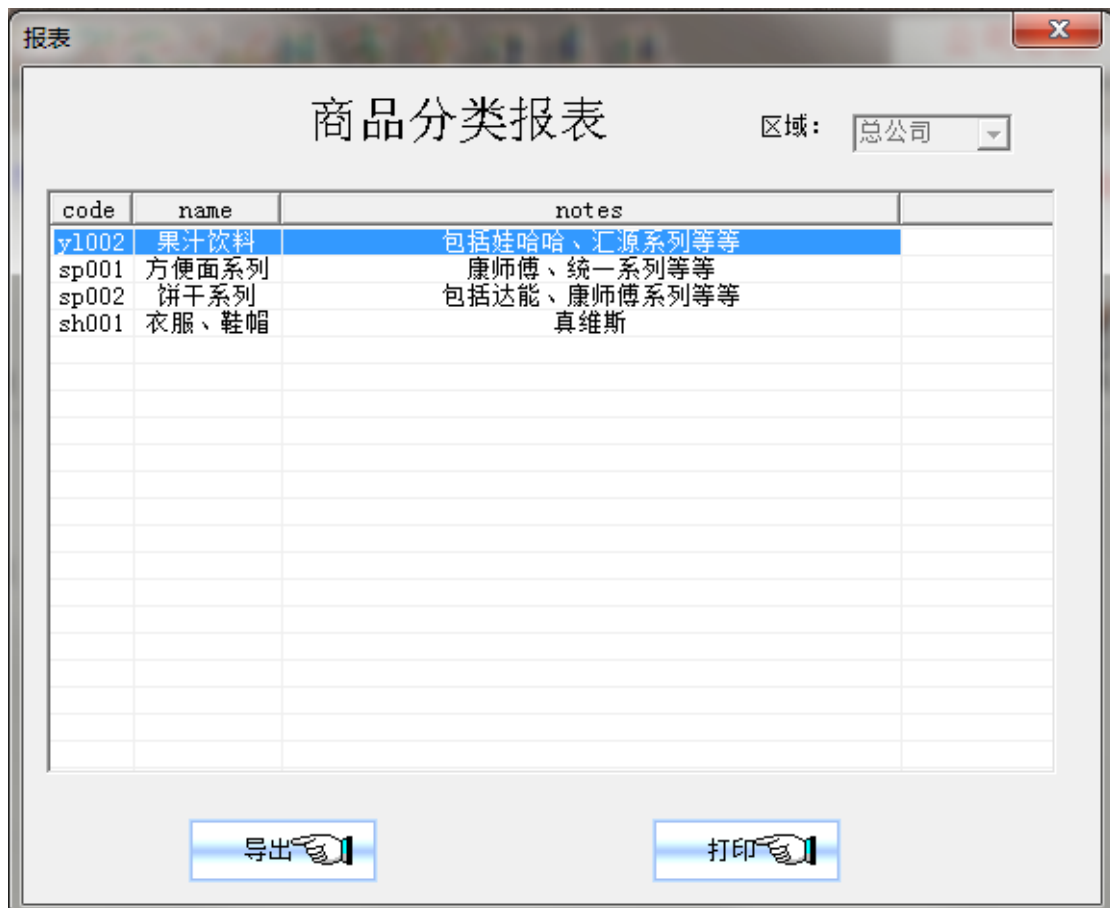


图 3-8 商品分类报表对话框

3.6 系统类库设计

BITVRLab 超市管理系统主框架设计是经过 MFC 创建向导创建基于对话框窗口程序，在对话框程序中添加了菜单栏，系统关键类库设计以下。

- 自定义扩展类: 为了提升开发效率、便于代码重用，自定义了部分数据操作类和控件扩展类，如表 3-1 所表示。

表 3-1 自定义扩展类及功效

类	说明
CADConn	经过 ADO 实现对 ACCESS、SqlServer 等数据库访问
CtablePartInfo	依据数据库表分配、分片信息，实现对数据库表增、删、改、查
CFileOperate	实现对硬盘文件常见操作
CMyChiToLetter	实现依据逐字提取汉字拼音首字母
CMyButton	CButton 类派生类，实现带有位图和文本按钮
CMyMenu	CMenu 类派生类，用于定制自己菜单
CMenuItemContext	CMenu 类中用到这类，用于保留菜单项信息
CMyTime	实现简单时间格式转换
CMyExcel	完成 VC 对 Excel 文件操作

- 对话框窗口类: 在系统中，用户全部数据查询、操作全部是经过对话框窗口来实现，系统开发对话框类如表 3-2 所表示。

表 3-2 对话框类及说明

类	说明
CDlgFenJiBase	POS 机基础信息管理对话框类
CShopManageDlg	主框架对话框类
CDlgFenLeiBase	商品分类基础信息管理对话框类
CDlgDanganBase	商品基础信息管理对话框类
CDlgGongYingShangBase	供给商基础信息管理对话框
GDlgJiaGeSheZhi	价格自动设置对话框类
CDlgJinHuoKanDan	商品进货开单管理对话框类

CDlgJinHuoTuiDan	商品进货退单管理对话框类
CDlgXiaoShouKanDan	商品销售开单管理对话框类
CDlgXiaoShouTuiDan	商品销售退单管理对话框类
CDlgAll	用于查找信息显示对话框类
CDlgReport	用于报表显示对话框类
CDlgLogo	用户登录对话框类
CDlgPwd	更改密码对话框类
CDlgQuit	退出系统提醒对话框类

➤ 打印相关类: 系统能够提供了报表打印和打印预览功效, 其相关设计类如表 3-3 所表示。

表 3-3 打印相关类及说明

类	说明
CPrintFrame	CFrameWnd 派生类, 用于构建打印框架类
CPrintView	打印视图类

另外, 系统还包含部分主框架相关类、导入 ActiveX 控件 (MFC Grid Control) 相关类和导入 OLE 对象 (Excel) 相关类。

3. 分布式数据库分析和设计

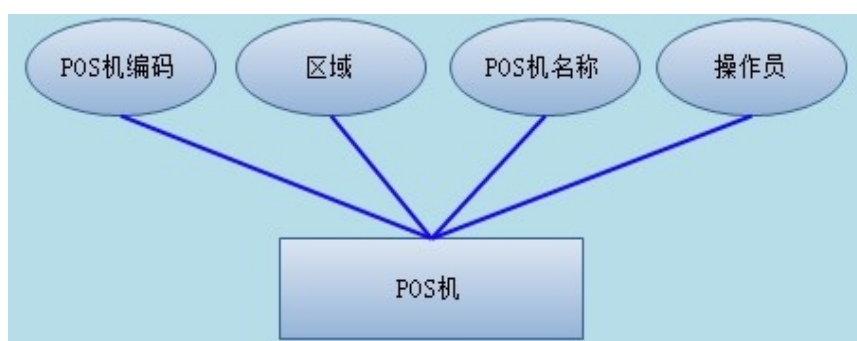
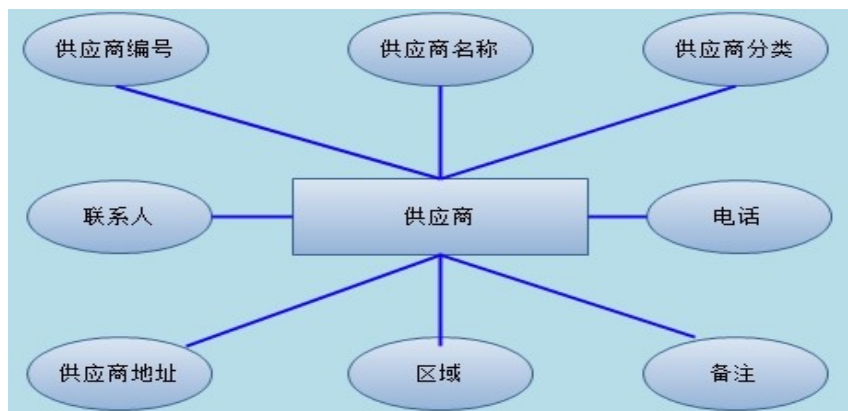
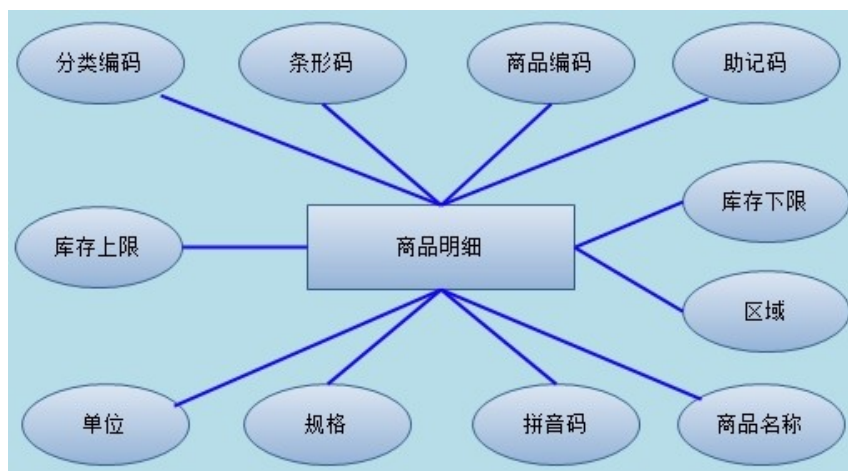
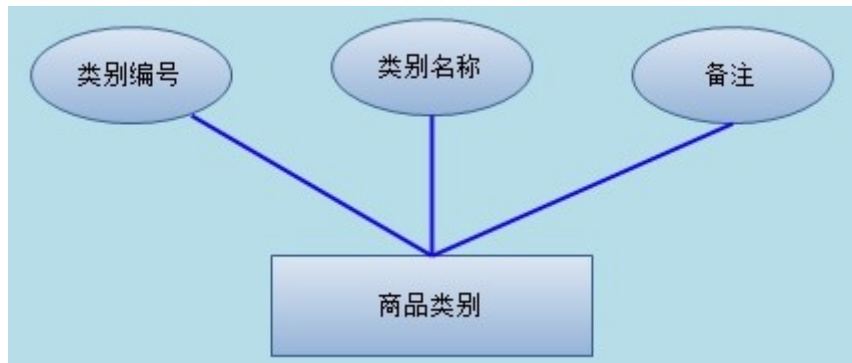
4.1 数据库分析

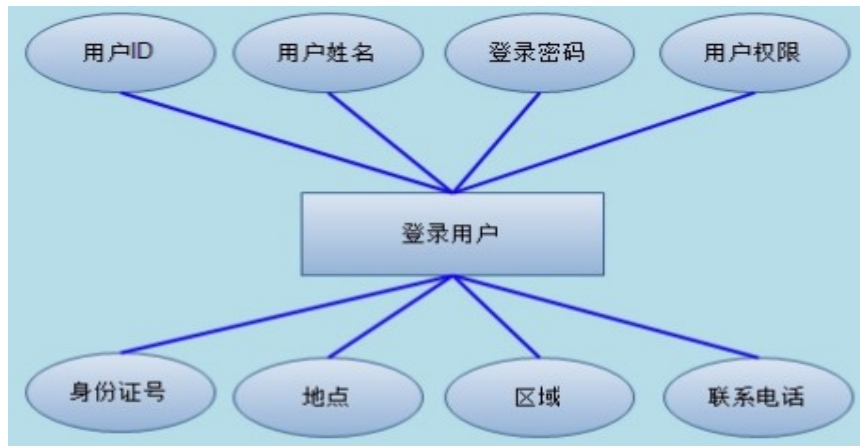
考虑到总站点信息量大采取 SQL Server 数据库, 而区域站点信息量相对较小采取 Access 数据库。当区域业务拓展, 操作终端增加时, Access 数据库也很轻易移植到 SQL Server 数据库系统中。另外区域站点采取 Access 数据库能够实现方便布署。

4.2 数据库概念设计

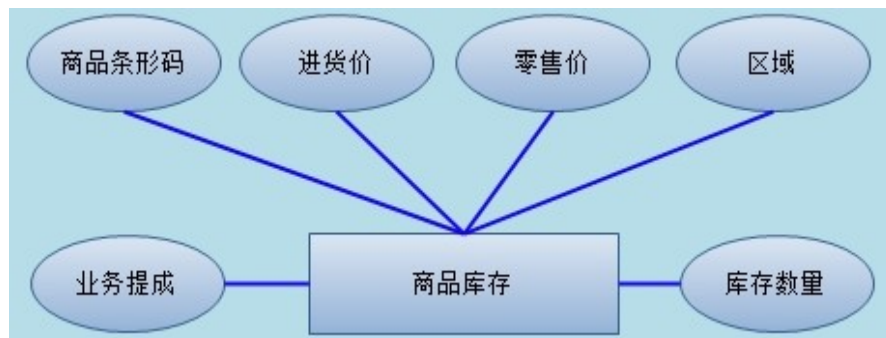
分析超市管理功效步骤, 系统数据实体关键包含基础资料对象实体、库存实体、进货/销售开单、进货/销售退单、进货/销售商品实体、各站点数据库服务器 IP 信息实体、数据库表分配分片信息实体等。

➤ 基础资料对象实体包含商品类别、商品明细、供货商、POS 机和登录用户实体。

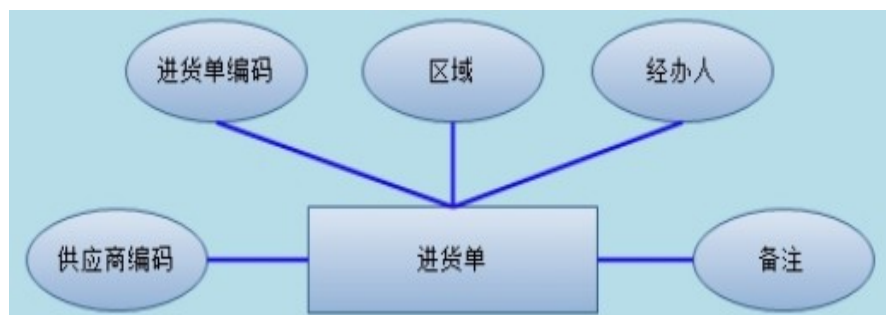




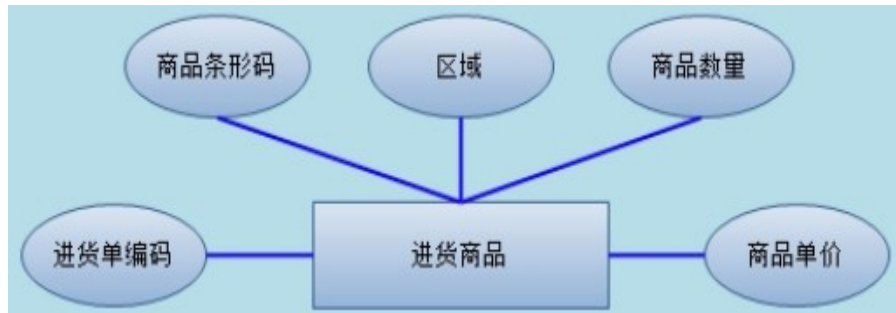
➤ 商品库存实体统计库存商品数量和价格信息,其实体 E-R 图以下。



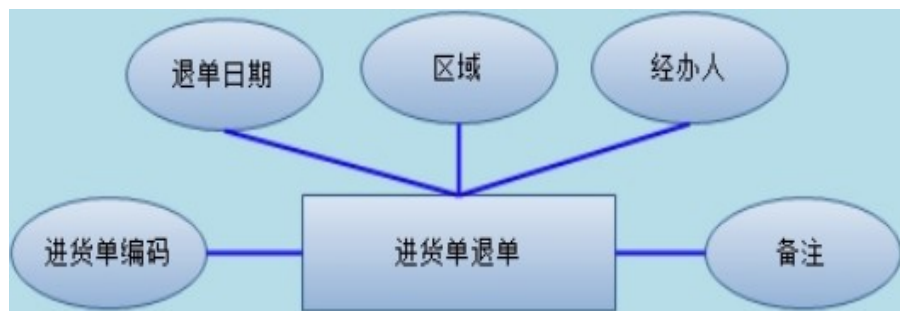
➤ 商品进货单实体用于统计进货单统计信息,其实体 E-R 图以下所表示。



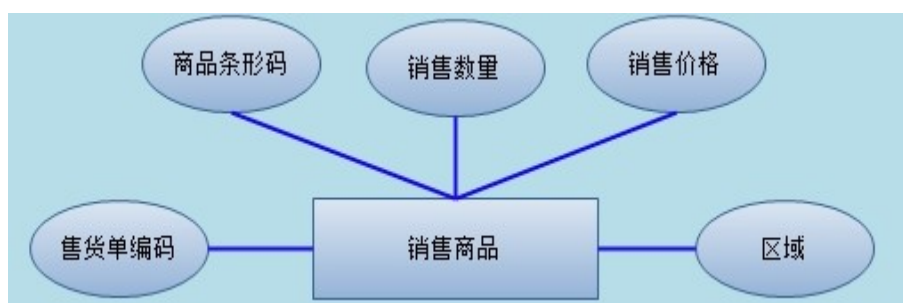
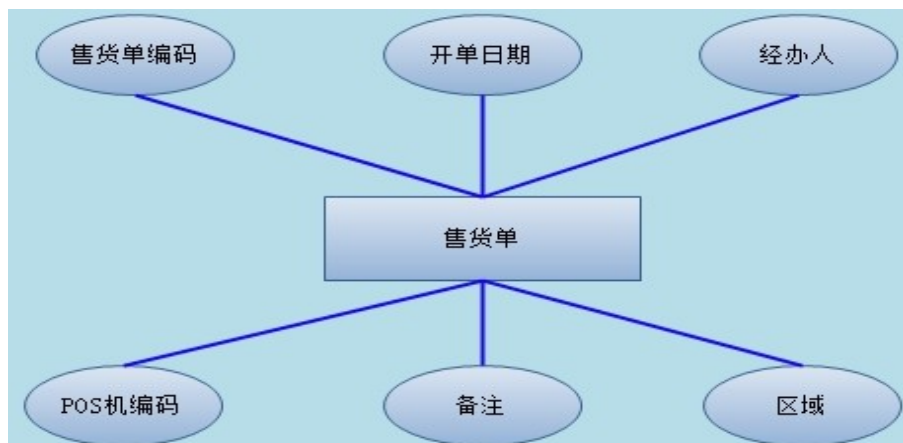
➤ 进货商品实体用于统计进货单对应商品信息,其实体 E-R 图以下所表示。

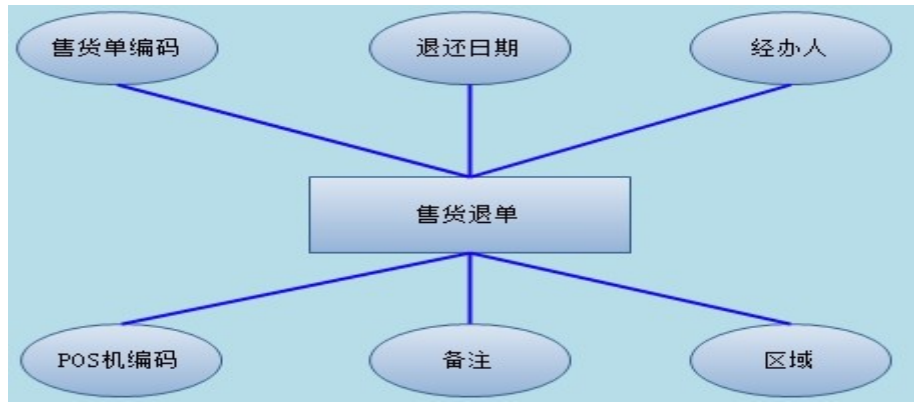


➤ 进货退单实体用于统计进货退单信息, 其实体 E-R 图以下所表示。

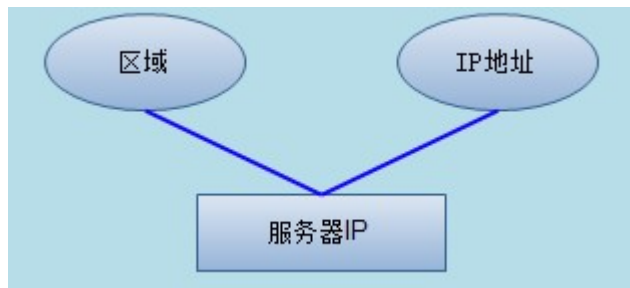


和进货相对应销售单实体、销售商品实体和销售退单实体 E-R 图以下所表示。

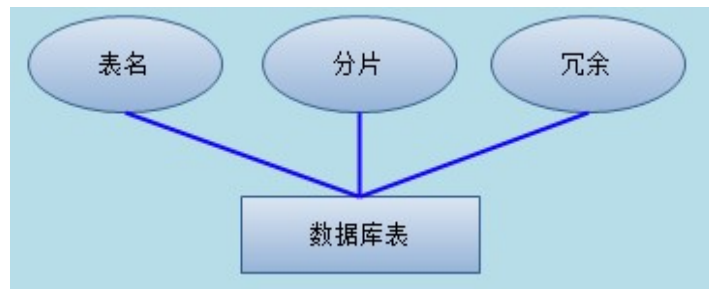




- 各站点数据库服务器 IP 信息，统计各个区域数据库服务器对应 IP 地址，其实体 E-R 图，以下图所表示。



- 数据库表分配分片信息，统计数据中全部表分布式分配和分片信息，其实体 E-R 图，图所表示。



4.3 数据库逻辑结构设计

商品类别表、商品明细表、供给商表、POS 机表、登录用户表、库存表、进货/销售开单表、进货/销售商品表、进货/销售退货商品表、IP 地址登录信息表、数据库表分配和分片信息。

商品类别表：

Table - dbo.CommodityType			
	Column Name	Data Type	Allow Nulls
	code	nvarchar(255)	<input type="checkbox"/>
	name	nvarchar(255)	<input checked="" type="checkbox"/>
	notes	nvarchar(255)	<input checked="" type="checkbox"/>

商品明细表:

Table - dbo.CommodityInfo			
	Column Name	Data Type	Allow Nulls
	typecode	nvarchar(255)	<input checked="" type="checkbox"/>
	barcode	nvarchar(50)	<input type="checkbox"/>
	commoditycode	nvarchar(50)	<input checked="" type="checkbox"/>
	memcode	nvarchar(50)	<input checked="" type="checkbox"/>
	spellcode	nvarchar(50)	<input checked="" type="checkbox"/>
	name	nvarchar(50)	<input checked="" type="checkbox"/>
	standard	nvarchar(50)	<input checked="" type="checkbox"/>
	unit	nvarchar(5)	<input checked="" type="checkbox"/>
	maxnum	float	<input checked="" type="checkbox"/>
	minnum	float	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>

供给商表:

Table - dbo.ProviderInfo			
	Column Name	Data Type	Allow Nulls
	code	nvarchar(50)	<input type="checkbox"/>
	name	nvarchar(50)	<input checked="" type="checkbox"/>
	sort	smallint	<input checked="" type="checkbox"/>
	contact	nvarchar(20)	<input checked="" type="checkbox"/>
	tel	nvarchar(20)	<input checked="" type="checkbox"/>
	address	nvarchar(255)	<input checked="" type="checkbox"/>
	notes	nvarchar(255)	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>

POS 机表:

Table - dbo.PosInfo			
	Column Name	Data Type	Allow Nulls
	code	nvarchar(50)	<input type="checkbox"/>
	name	nvarchar(50)	<input checked="" type="checkbox"/>
	inchage	nvarchar(255)	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>

用户表:

Table - dbo.AdminInfo			
	Column Name	Data Type	Allow Nulls
	code	nvarchar(50)	<input type="checkbox"/>
	name	nvarchar(20)	<input checked="" type="checkbox"/>
	authority	smallint	<input checked="" type="checkbox"/>
	IDcard	nvarchar(20)	<input checked="" type="checkbox"/>
	pwd	nvarchar(255)	<input checked="" type="checkbox"/>
	tel	nvarchar(20)	<input checked="" type="checkbox"/>
	address	nvarchar(255)	<input checked="" type="checkbox"/>
	area	int	<input checked="" type="checkbox"/>



库存表:

Table - dbo.StoreInfo			
	Column Name	Data Type	Allow Nulls
	barcode	nvarchar(50)	<input type="checkbox"/>
	stockprice	float	<input checked="" type="checkbox"/>
	saleprice	float	<input checked="" type="checkbox"/>
	insiderprice	float	<input checked="" type="checkbox"/>
	bonus	float	<input checked="" type="checkbox"/>
	storenum	smallint	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>



进货单表:

Table - dbo.StockBill			
	Column Name	Data Type	Allow Nulls
	stockcode	nvarchar(50)	<input type="checkbox"/>
	stockdate	nvarchar(20)	<input checked="" type="checkbox"/>
	transactor	nvarchar(50)	<input checked="" type="checkbox"/>
	providercode	nvarchar(50)	<input checked="" type="checkbox"/>
	notes	nvarchar(255)	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>



进货商品表:

Table - dbo.StockCommodity			
	Column Name	Data Type	Allow Nulls
	stockcode	nvarchar(50)	<input type="checkbox"/>
	barcode	nvarchar(50)	<input checked="" type="checkbox"/>
	stockprice	float	<input checked="" type="checkbox"/>
	stocknum	smallint	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>



进货商品退单表:

Table - dbo.StockReturn			
	Column Name	Data Type	Allow Nulls
	stockcode	nvarchar(50)	<input type="checkbox"/>
	returndate	nvarchar(20)	<input checked="" type="checkbox"/>
	transactor	nvarchar(50)	<input checked="" type="checkbox"/>
	notes	nvarchar(255)	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>

销售单表:

Table - dbo.SaleBill			
	Column Name	Data Type	Allow Nulls
	salecode	nvarchar(50)	<input type="checkbox"/>
	saledate	nvarchar(20)	<input checked="" type="checkbox"/>
	transactor	nvarchar(50)	<input checked="" type="checkbox"/>
	poscode	nvarchar(50)	<input checked="" type="checkbox"/>
	notes	nvarchar(255)	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>

销售商品表:

Table - dbo.SaleCommodity			
	Column Name	Data Type	Allow Nulls
	salecode	nvarchar(50)	<input type="checkbox"/>
	barcode	nvarchar(50)	<input checked="" type="checkbox"/>
	saleprice	float	<input checked="" type="checkbox"/>
	salenum	float	<input checked="" type="checkbox"/>
	area	int	<input type="checkbox"/>

销售商品退单表:

Table - dbo.SaleReturn			
	Column Name	Data Type	Allow Nulls
🔑	salecode	nvarchar(50)	<input type="checkbox"/>
	returndate	nvarchar(20)	<input checked="" type="checkbox"/>
	transactor	nvarchar(50)	<input checked="" type="checkbox"/>
	notes	nvarchar(255)	<input checked="" type="checkbox"/>
	postcode	nvarchar(50)	<input checked="" type="checkbox"/>
🔑	area	int	<input type="checkbox"/>

服务器 IP 信息表:

Table - dbo.IPInfo			
	Column Name	Data Type	Allow Nulls
🔑	area	int	<input type="checkbox"/>
	IPAddress	nvarchar(50)	<input checked="" type="checkbox"/>

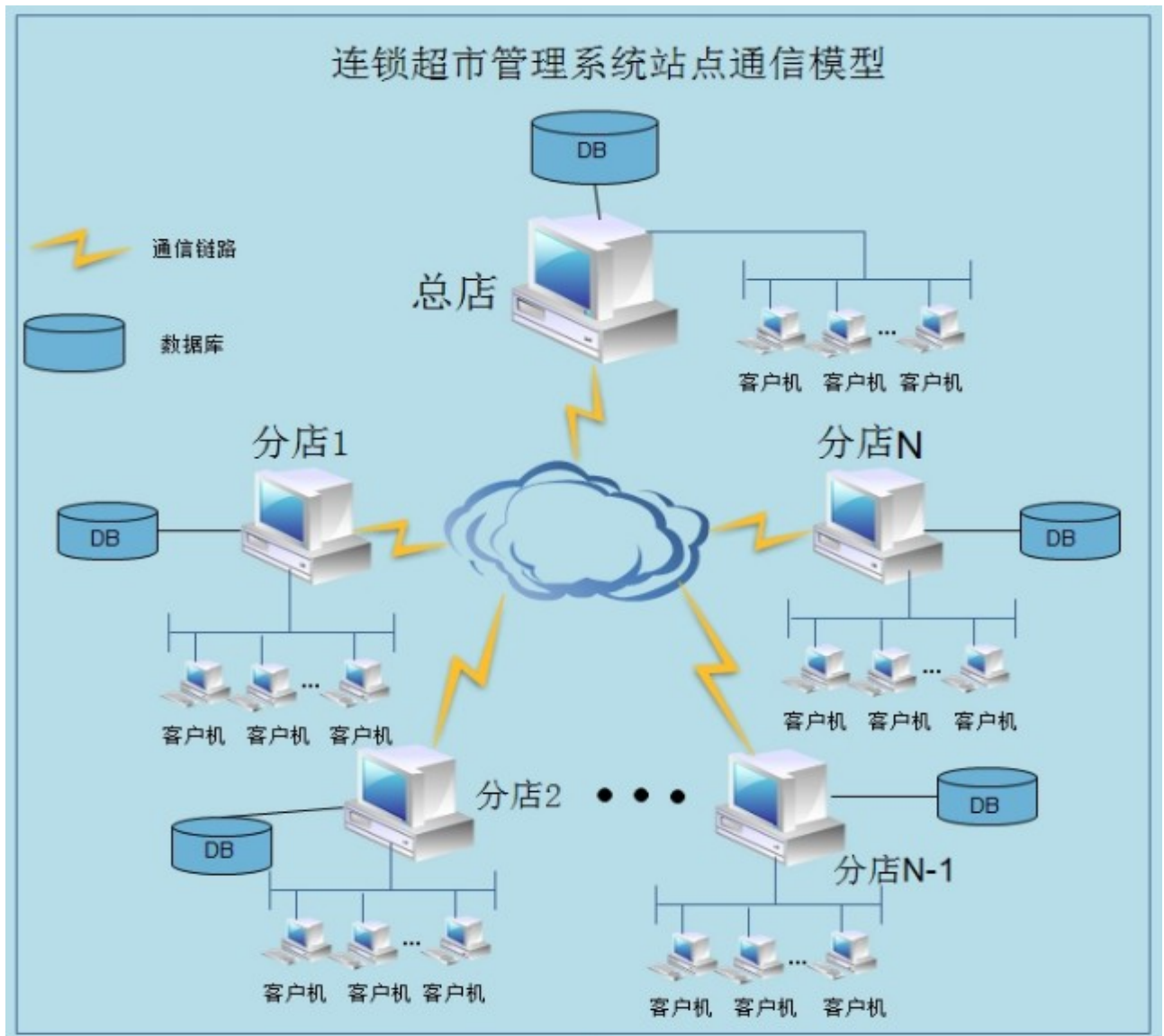
分片及分布信息表:

Table - dbo.TablePartInfo			
	Column Name	Data Type	Allow Nulls
🔑	TableName	nvarchar(50)	<input type="checkbox"/>
	Part	int	<input checked="" type="checkbox"/>
	Copy	int	<input checked="" type="checkbox"/>

4.4 分片和位置分配设计

4.4.1 站点通信模型

在我们连锁超市管理系统中，有一个总店并下分多个分店，总店和分店之间或分店和分店之间全部能够进行通信。每个分店是一个相对独立数据库服务系统，其能够连接任意数量客户端。通信模型以下图所表示：



4.4.2 数据表分片和位置分配设计

依据以上得到相关关键信息数据需求分析结果，为该连锁超市系统分布式数据库系统进行关键信息数据分片和分配设计以下：

1、数据分片设计

(1) 因为该连锁超市系统各连锁店之间在经营上是独立，每个门店只关心自己经营情况，相关供给商信息、POS 机信息、商品信息和进货/销货信息等基础数据全部是各门店单独管理和使用，门店之间互不相关。所以，商品明细表、供给商表、POS 机表、库存表、进货/销售开单表、进货/销售商品表、进货/销售退货商品表根据地域（门店所在区域标志）采取水平分片方法得到水平片段。

(2) 这里我们对于商品信息进行了垂直分片，分成了商品明细表和库存表，因为商品一些属性，比如库存量等需要常常更新，所以将这些属性划分出来组成单独实体能够降低系统开销。

(3) 因为整个连锁超市职员信息由总企业管理和维护，各门店只能够查询本店职员信息。所以职员信息无须分片，能够采取视图形式提供给各门店查询本门店职员信息。另外，各个区域数据库服务器 IP 地址信息也只是由总部管理、维护，所以也无须分片。

(4) 因为商品分类数据由总部统一管理并提供各门店使用，而且，商品类别信息数据在各门店全部要常常使用。所以，商品类别信息数据也无须分片。

(5) 该分布式数据库系统实现了简单目录管理，统计数据库中各个表分片和分配信息，方便数据更新时，维护各个站点上数据一致性。这个表由总部计划建立，各门店也会常常使用。所以，数据库目录信息表也无须分片。

2、数据及其片段分配设计

(1) 对于只在各门店单独使用除商品类别信息和数据库目录信

息以外其它基础信息片段,采取按区域分片然后分配到各个门店数据库服务器上。总站上有全部门店全部信息。

(2) 整个系统职员信息、IP 地址信息由企业总部管理和维护，所以只分配在总部站点中。

(3) 商品类别信息、数据库目录信息是由总部统一要求并下发到各门店，因为各店常常会使用到这类基础信息，所以在各门店全部含有相同副本。所以，商品类别信息、数据库目录信息全部不会分片但被复制，且复制个数为门店个数。

4. 基于 SQLSMS 具体设计（实现）

5.1 公共类设计

为了提升程序代码开发效率，便于代码重用，在系统开发中，创建了部分数据操作类和控件扩展类。

5.1.1 文件操作类 CFileOperate

为了便于对硬盘文件操作，开发了文件操作类 CFileOperate，它经过调用 API 函数实现常见文件操作。CFileOperate 类申明代码如下。

```
extern CString strTmpPath;

class CFileOperate
{
public:
    //结构函数
    CFileOperate();

    //取得目前途序运行路径
    CString GetAppPath();

    //判定是否存在 strFn 文件夹
```

```
BOOL IsFileExist(CString strFn, BOOL bDir);
```

```
//制作 strFloderName 文件夹返回文件夹名
```



```
CString MakeDirectory(CString strFolderName);  
  
//自动生成文件夹  
  
CString MakeDirectory();  
  
//得到文件夹名为 strFolderName 路径  
  
CString GetDirectoryPath(CString strFolderName);  
  
//删除 strFolderName 文件夹  
  
void delDirectory(CString strFolderName);  
  
//删除主文件夹(data)  
  
void delMainDirectory();  
  
//制作主文件夹  
  
void MakeMainDirectory();  
  
//判定 strIntDigit 是否为整数，是否小于 intBig  
  
BOOL CheckIntDigit(CString strIntDigit,int intBig);  
  
//判定 strFileName 是否能够做文件夹或文件名字  
  
BOOL CheckFileName(CString strFileName);  
  
//判定 strText 是否为空  
  
BOOL CheckEmpty(CString strText);  
  
//删除 strFolderName 文件夹下名为 strFileName 文件  
  
void delFile(CString strFolderName,CString strFileName);  
  
//得到 strFolderName 文件夹下名为 strFileName 文件路径  
  
CString GetFileName(CString strFolderName,CString strFileName);  
  
//制作 strFolderName 文件夹下名为 strFileName 文件  
  
CString MakeFile(CString strFolderName,BOOL bInMake);  
  
//显示文件夹对话框  
  
BOOL GetFolder(CString* strSelectedFolder,const char* lpszTitle,const HWND  
hwndOwner,const char* strRootFolder,const char* strStartFolder);
```

系统程序中关键用到了 CFileOperate 类中 GetAppPath 函数。GetAppPath 函数用于获取目前运行程序所在文件夹路径，其实现代码以下。

```
CString CFileOperate::GetAppPath()//取得目前运行程序所在文件夹路径
{
    char lpFileName[MAX_PATH];//路径数组
    //获取目前运行程序全路径
    GetModuleFileName(AfxGetInstanceHandle(),lpFileName,MAX_PATH);
    CString strFileName = lpFileName;
    //从字符串最右边向左搜索'\'串
    int nIndex = strFileName.ReverseFind ('\\');
    CString strPath;
    if (nIndex > 0)
        strPath = strFileName.Left (nIndex);//取'\'串左边字符
    else
        strPath = "";
    return strPath;//返回前运行程序文件夹路径
}
```

5.1.2 时间格式转换类 CMyTime

系统开发中，需要频繁对日期、时间数据进行操作，所以这里设计了类 CMyTime，实现简单时间日期转换，CMyTime 类申明以下。

```
class CMyTime
{
public:
    CTime ValueTime;//CTime 类型值
```

```
//blnChinese:TRUE-"年月日时分秒" FALSE-"-:"
```

```
CString GetAllString(BOOL blnChinese); //返回字符串形式日期时间
```

```

//blnChinese:TRUE-"年月日" FALSE-"-"
CString GetDateStr(CString Value,BOOL blnChinese);//返回字符串形式日期

//blnChinese:TRUE-"时分秒" FALSE-":":
CString GetTimeString(CString Value,BOOL blnChinese);//返回字符串形式时间

CString GetSimpleString();//返回简单字符串形式日期时间

CString GetWeek();//返回星期"星期日"

//设置字符串形式值

//blnSimple:TRUE-(%Y%m%d%H%M%S) FALSE-(年月日时分秒或-)

void SetAllString(CString Value,BOOL blnSimple);

void SetNow();//设置成现在时间

CMyTime();//构造函数

};

```

SetNow 函数获取目前日期，时间，将其值给予 ValueTime。

```

void CMyTime::SetNow()
{
    ValueTime=CTime::GetCurrentTime();
}

```

5.1.3 数据操作类 CADConn

系统开发了数据操作类 CADConn，它是本系统关键，实现了对数据库数据基础操作功效，CADConn 类申明代码如下。

```

#import "c:\program files\common files\system\ado\msado15.dll" \
    no_namespace rename ("EOF", "adoEOF")

class CADConn
{
// 定义变量

public:

```

```

_ConnectionPtr m_pConnection;//指向 Connection 对象指针:

//添加一个指向 Recordset 对象指针:

_RecordsetPtr m_pRecordset;

_bstr_t m_strData;

int m_DataType;

CADOConn(int DataType);//构造函数

CADOConn();

virtual ~CADOConn();

void OnInitCADOConn();//初始化连接数据库

_RecordsetPtr& GetRecordSet(CString strSQL);//实施查询

BOOL ExecuteSQL(CString strSQL);//实施 SQL 语句, Insert Update delete

void ExitConnect();//退出连接

BOOL MoveFirst();//字段集移向开头

BOOL MoveNext();//字段集向下移

BOOL Open(CString strSQL);//打开统计集

BOOL OpenLogo(CString strSQL);//打开统计集

CString GetValueString(int index,int strSum);//返回统计集中某字段字符串

byte GetValueByte(int index);//返回统计集中某字段字节

int GetValueInt(int index);//返回统计集中某字段短整数

double GetValueDouble(int index);//返回统计集中某字段双精度数

float GetValueFloat(int index);//返回统计集中某字段单精度数

long GetValueLong(int index);//返回统计集中某字段长整型数

CTime GetValueDate(int index);//返回统计集中某字段日期时间

//获取统计集某字段 BYTE 值, 并换为 CString 返回

CString GetValueByteStr(int index,int strSum);

//获取统计集某字段 INT 值, 并换为 CString 返回

CString GetValueIntStr(int index,int strSum);

//获取统计集某字段 Double 值, 并换为 CString 返回

```

```

CString GetValueDoubleStr(int index,int strLSum,int strRSum);
//获取统计集某字段 Float 值，并换为 CString 返回

CString GetValueFloatStr(int index,int strLSum,int strRSum);
//获取统计集某字段 Long 值，并换为 CString 返回

CString GetValueLongStr(int index,int strSum);
//获取统计集某字段 CTime 值，并换为 CString 返回

CString GetValueDateStr(int index,CString strType);
//添加单项数据

BOOL AddItem(CString strTable,int strSum,LPCTSTR pszText, ... );
//得到字段中数据类型

int GetValueType(int index);

BOOL adoEOF();//统计集结束判定

BOOL FillList(CListCtrl *listMain,int ColOpenEnd);//填充列表(ColOpenEnd 代表展开多少列)

BOOL InitList(CListCtrl *listMain,int colSum);//初始化列表

CString GetAppPath();//得到应用程序所在文件夹

BOOL FillList(CListCtrl *listMain);//填充列表

CString GetFieldsName(int index);//返回字段名字

int GetFeildsCount();//返回字段数量

//返回数据集数

long GetRecordCount();

void WriteLog(CString strSql);//写日志文件

void WriteLog1(CString userName);//写日志文件,谁谁登录
};

```

1、数据库操作函数

对常见数据库操作定义了相关函数。

(1) 连接数据库

在类构造函数 CADOConn 中，申明数据库名称。

```
CADOConn::CADOConn()
{
    m_DataType=1;//数据库类型 Access
    m_strData=_bstr_t("ShopData.mdb");//数据库名称
}
```

在 OnInitCADOConn 函数中，实现连接数据库。

```
void CADOConn::OnInitCADOConn()
{
    // 初始化 OLE/COM 库环境
    ::CoInitialize(NULL);
    try
    {
        //初始化指针
        m_pConnection=NULL;
        //初始化指针
        m_pRecordset=NULL;
        // 创建 Connection 对象
        m_pConnection.CreateInstance("ADODB.Connection");
        // 设置连接字符串，必需是 BSTR 型或_bstr_t 类型
        _bstr_t strConnect;
        switch(m_DataType)
        {
            case 1://ACCESS
                strConnect=_bstr_t("Provider=Microsoft.Jet.OLEDB.4.0;");
                strConnect=strConnect+_bstr_t("Data Source=");
                strConnect=strConnect+_bstr_t(IPAddress)+_bstr_t("\\data\\");
            }
    }
}
```

```

        strConnect=strConnect+m_strData;

        break;

    case 2://EXCEL

        strConnect=_bstr_t("Provider=Microsoft.Jet.OLEDB.4.0;");

        strConnect=strConnect+_bstr_t("Data Source=");

        strConnect=strConnect+_bstr_t(GetAppPath()+_bstr_t("\\"));

        strConnect=strConnect+m_strData;

        strConnect=strConnect+";Extended Properties=Excel 8.0";

        break;

    case 3://SQLSERVER

        strConnect="Provider=SQLOLEDB; Server="+MainIP+";Database=ShopData;

uid=admin; pwd=123456";

        break;

    }

    m_pConnection->Open(strConnect,"", "",adModeUnknown);

}

// 捕捉异常

catch(_com_error e)

{

    // 显示错误信息

    AfxMessageBox(e.Description());

}

ASSERT(m_pConnection != NULL);

}

```

(2) 对数据库进行查询

GetRecordSet 函数实现实施 Select 查询语句，返回查询结果集

```

_RecordsetPtr& CADConn::GetRecordSet(CString strSQL)

```


以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/498142054123006141>