

# 陕西科技大学



## 题目：基于大型连锁超市管理系统的分布式数据库设计

姓 名： Deran Pan

学 号：

学 院： 电气与信息工程学院

专 业： 计算机技术

2018 年 6 月 20 日



## 目录

1、	背景介绍 .....	1
2、	需求分析 .....	2
2.1、	用户需求概述 .....	2
2.2、	功能分析 .....	2
3、	分布式数据库的设计 .....	4
3.1、	数据库分析 .....	4
3.2、	总体设计 .....	4
3.3、	概念设计 .....	6
3.4、	数据库逻辑结构设计 .....	8
3.5、	分片与位置分配设计 .....	12
4、	分布式数据库查询优化策略； .....	14
4.1、	基本概念 .....	14
4.2、	用于连锁超市的分布式数据库的查询优化 .....	14
4.2.1、	基于关系代数等价变换的优化算法 .....	14
4.2.2、	基于半连接操作的查询优化算法 .....	16
5、	事务处理方式； .....	17
5.1、	基本概念 .....	17
5.2、	分布式事务处理流程 .....	17
6、	并发控制处理机制 .....	19
6.1、	基本概念 .....	19
6.2、	并发控制方法 .....	19
6.2.1、	两阶段封锁机制(2PL) .....	19
6.2.2、	其他并发控制协议 .....	20
7、	分布式数据库系统的安全性、可靠性 .....	21
7.1、	安全性 .....	21
7.2、	可靠性 .....	21



## 1、 背景介绍

随着市场经济的发展和人民生活水平的提高，原来单一、小规模超市已无法满足人民对购物环境的要求，大规模、物品丰富的超市正在蓬勃发展。连锁超市的出现更是使得连锁超市由大到多的转变，使得他们具有地域上分散而管理上又相对集中的特点，往往既要有各门店的局部控制和分散管理，同时也要有整个组织的全局控制和高层次的协同管理。

超市销售数据规模的日益庞大，商品数目的迅速增长，采用以往的手工管理已直接或间接地降低了工作效率，由于无法及时掌握最新的各连锁店的各项事实数据，导致决策的延迟和滞后，甚至管理上的混乱，以至于无法对商品进行及时有效的调配和流转，造成无法计算的损失。最终影响超市的日常运转。

另外超市的扩张壮大，尤其是连锁超市（如家乐福、美廉美）的扩张因此使用连锁超市管理系统将这些门店和中心通过网络管理起来则显得势在必行。而该套管理系统又势必基于分布式数据库系统，因此设计一个适当的分布式数据则显得十分必要。

## 2、 需求分析

### 2.1、 用户需求概述

下面是通过调查研究获得的关于连锁超市主要信息数据的需求分析结果：

- 1) 连锁超市通常由一个中心（公司总部）、多个远程连锁店（门店）组成，并且每个连锁店分布在不同地域。
- 2) 连锁超市的各个部门之间、各个分店之间、分店与总部之间需要交换数据，这种数据交换是通过局域网和广域网进行的。
- 3) 公司总部负责产生并管理该连锁超市的整体汇总数据，即各门店的明细汇总表数据，如销售汇总表等。
- 4) 每一个远程站点（各门店和公司总部）分别有一个数据库系统，各自组成一个独立的子系统，可以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的销售情况，要求门店将所有商品归入相应的商品类别，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都要经常使用。
- 6) 有关商品信息、供应商信息、进货信息和销售信息等经营基础数据都是各门店单独管理和使用，门店之间互不相关。
- 7) 整个连锁超市的职员信息由公司总部管理和维护，各门店只可以查询本部门的职员信息。

### 2.2、 功能分析

根据对系统的业务调查和用户的需求分析，管理系统需要实现的功能如下。

- 1) **分店管理**：分店管理主要实现对连锁店的信息管理，包括分店店名，地址，负责人等，其他登录用户（职员）管理、系统日志、修改登录密码等相关工作。
- 2) **商品信息管理**：基础信息管理包括业种商品类别信息、商品信息、价格等信息的维护与管理，实现的功能包括基本信息的添加、删除和更新操作、生成各类基础信息报表、打印和导出报表。
- 3) **进货管理**：进货管理主要实现对商品的采购入库信息进行管理，具体实现功能有进货

开单，实现商品的进货结算、入库操作；进货退货，退还商家相关的商品；生成进货、退货的单据、商品报表；打印和导出报表。

- 4) **销售管理：**销售管理主要实现对商品的销售出库相关信息进行管理，具体实现的功能有销售开单，实现商品的销售结算、出库操作；销售退货，允许客户退出相关的商品；生成销售、销售退货的单据、商品报表；打印和导出报表。
- 5) **库存管理：**库存管理主要实现对商品的库存相关信息进行管理，具体实现的功能有库存查询，可以查看所有库存商品的相关信息；库存报警，对库存过多或过少的商品进行报表统计；库存盘点，可以修改商品的库存数量。
- 6) **财务管理：**财务管理主要实现对营业员的销售商品、业务提成、营业收入等情况进行分类报表统计。
- 7) **终端管理：**终端管理主要实现对所在分店终端的信息与授权管理等相关工作。

### 3、 分布式数据库的设计

#### 3.1、 数据库分析

考虑到连锁超市总店的信息量大，故而采用 SQL Server 2005 数据库，而较小的连锁分店信息量相对较小采用 Access 数据库。当连锁店业务拓展操作终端增加时，Access 数据库也很容易的移植到 SQL Server 数据库系统中。另外在连锁店点采用 Access 数据库可以实现方便的部署。

#### 3.2、 总体设计

本系统将用到分布式数据库查询。传统数据库随着应用的不断发展，规模的不断扩大，逐渐使得集中式系统越来越不适合与在大型连锁超市中应用。而大型分布式 DBS(Data Base System) 的设计和操作都是比较复杂的。DDBS(Distributed Data Base System) 是物理上分散，逻辑上集中的数据库系统。系统中的数据分布存放在计算机网络的不同场地的计算机中，每一场地都有自治处理（即独立处理）能力并能完成局部应用，而每一场地也参与（至少一种）全局应用，程序通过网络通信子系统执行全局应用。

本应用于连锁超市管理系统的分布式数据库总体结构如图 1 所示。



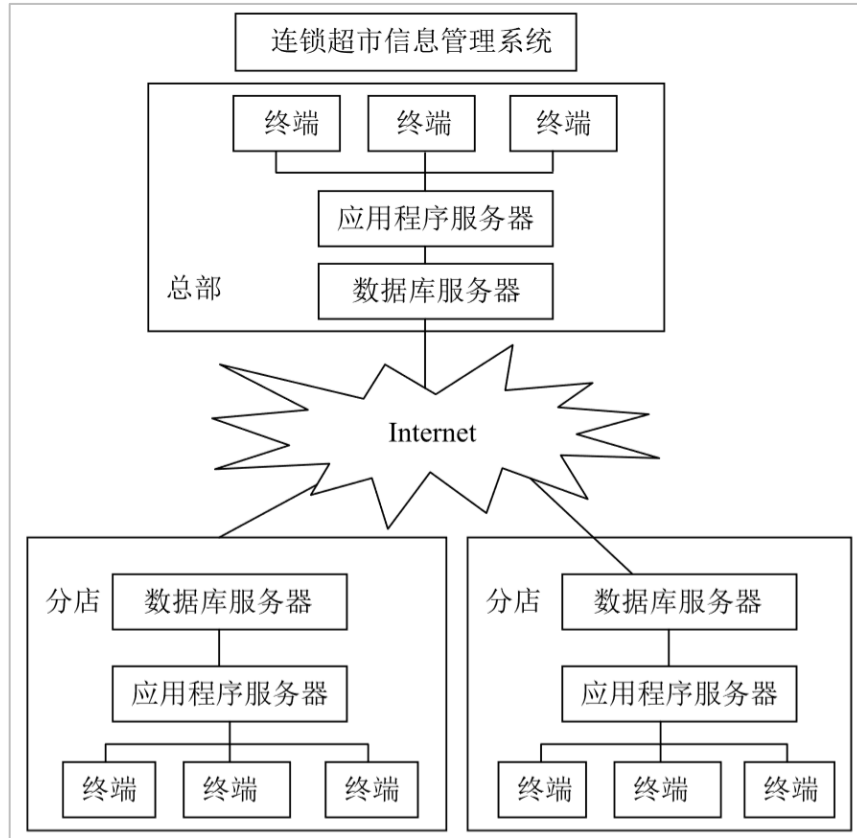


图 1

本应用于连锁超市管理系统的分布式数据库的组成框架如图 2 所示。

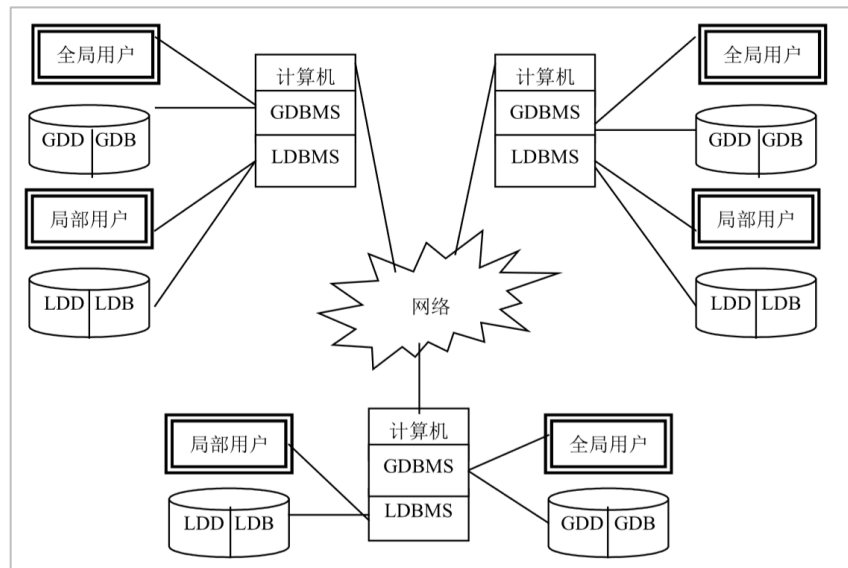


图 2

3.3、 概念设计

在本系统中，每种商品可以由多个销售人员销售，每名销售人员也可以负责销售多种商品，一个销售人员只能在一个终端销售商品，一个终端也只能由一个销售人员操作，每个供应商可以供应多种商品，每种商品可以由多个供应商供应，一个入库人员可以对多种商品入库，同种商品也可以被多个入库人员入库，商品、销售人员、终端、供应商从属于一间分店，分店与上述实体之间是一对多关系。根据上面描述，可以构建如下的 E-R 图。

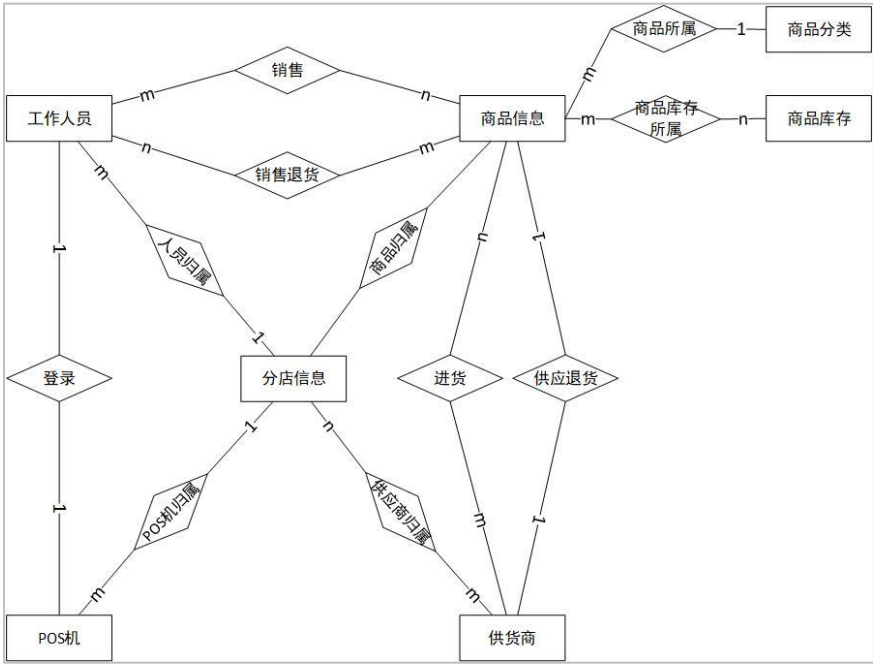
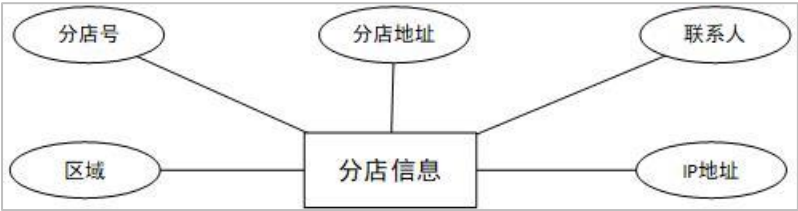


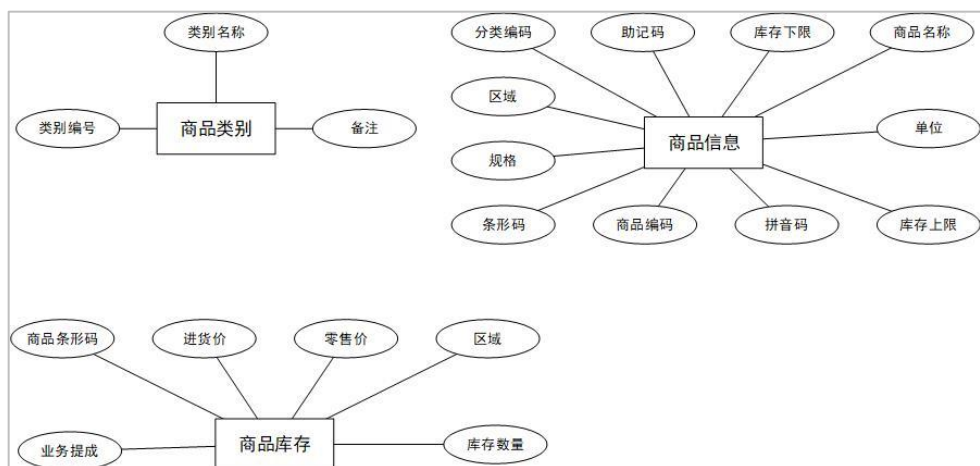
图 3

分析超市管理功能流程，结合分布式数据的特点以及上述 ER 图。系统的数据实体以及弱实体主要包括分店对象实体、商品实体、供应商实体、工作人员实体，POS 实体，进货/供应退货弱实体、销售/销售退货弱实体、登录弱实体以及相关归属弱实体等。

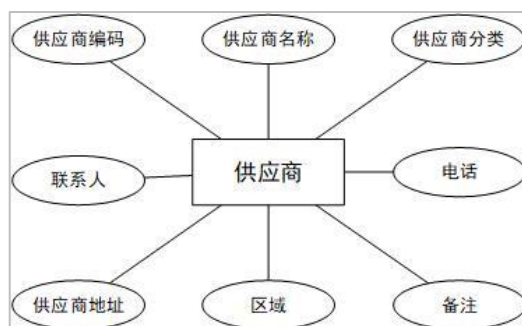
分店对象实体分店号、分店地址、联系人、区域、IP 地址等。



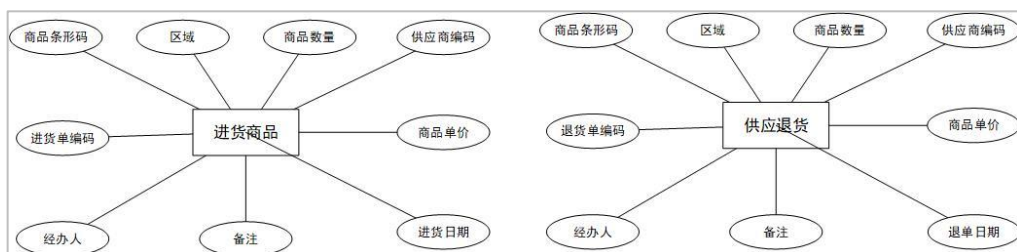
商品信息实体记录商品的信息和价格，其实体的如下图所示。



供应商实体用于记录供应商的信息，其实体的如下图所示。



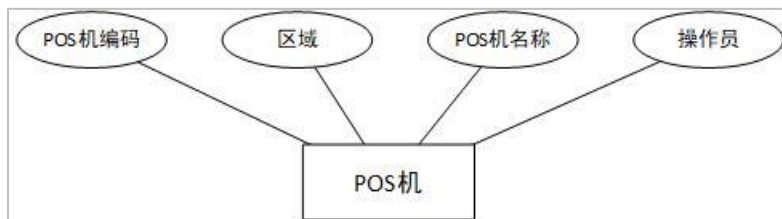
进货/退货实体用于记录进货/退货单对应的商品信息，其实体的如下图所示。



销售/销售退货单实体用于记录销售和其退单信息，其实体的如下图所示。



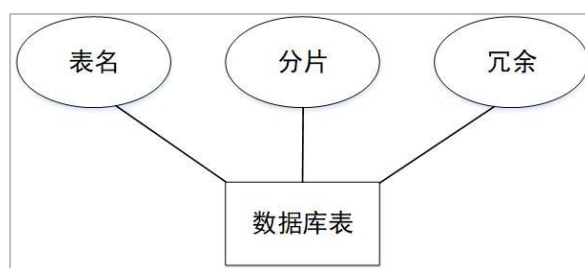
POS 机的实体信息的如下图所示。



工作人员主要用于管理人员的信息，其实体的如下图所示。



数据库表分配分片信息，记录数据中所有的表的分布式分配和分片信息，其实体 E-R 图，如图所示。



### 3.4、 数据库逻辑结构设计

根据上述章节所述，本应用的连锁超市的数据库的逻辑结构应包含商品类别表、商品表、供应商表、库存表、进货表、供应退货表、登录表、销售表、销售退货表、分店信息表以及数据库表的分配和分片信息等。

#### 分店信息表 (branch\_supermarket\_Infomation)

字段	类型	说明
id	nvarchar (50)	分店号
addr	nvarchar (50)	分店地址
manager	nvarchar (50)	联系人
ip_addr	nvarchar (50)	IP 地址

<b>region</b>	nvarchar (50)	区域
---------------	---------------	----

**商品类别表 (Prdt\_Classfication)**

字段	类型	说明
<b>id</b>	nvarchar (50)	类别编号
<b>name</b>	nvarchar (50)	类别名称
<b>comment</b>	nvarchar (256)	备注

**商品表 (Prdt\_Detail\_Infomation)**

字段	类型	说明
<b>clf_id</b>	nvarchar (50)	分类编码
<b>mem_code</b>	nvarchar (50)	助记码
<b>stock_floor</b>	Int	库存下限
<b>prdt_name</b>	nvarchar (50)	商品名称
<b>prdt_unit</b>	nvarchar (50)	单位
<b>stock_upper</b>	Int	库存上限
<b>pinyin_code</b>	nvarchar (50)	拼音码
<b>prdt_id</b>	nvarchar (50)	商品编码
<b>prdt_bar_code</b>	nvarchar (50)	条形码
<b>prdt_spec</b>	nvarchar (50)	规格
<b>prdt_region</b>	nvarchar (50)	区域

**商品库存表 (Prdt\_Inventory)**

字段	类型	说明
<b>prdt_bar_code</b>	nvarchar (50)	商品条形码
<b>purchase_price</b>	decimal (9, 2)	进货价
<b>retail_price</b>	decimal (9, 2)	零售价
<b>region</b>	nvarchar (50)	区域
<b>prdt_stock_cnt</b>	Int	库存数量

**供应商表 (Suppliers)**

字段	类型	说明
code	nvarchar (50)	供应商编码
name	nvarchar (50)	供应商名称
classification	nvarchar (50)	供应商分类
phone_name	nvarchar (50)	电话
region	nvarchar (50)	区域
addr	nvarchar (50)	地址
contacts	nvarchar (50)	联系人
comment	nvarchar (50)	备注

进货商品表（Prdt\_Stock）

字段	类型	说明
bar_code	nvarchar (50)	商品条形码
region	nvarchar (50)	区域
cnt	Int	商品数量
supplier_code	nvarchar (50)	供应商编码
unit_price	decimal (9, 2)	商品单价
date	nvarchar (50)	进货日期
operator	nvarchar (50)	经办人
order_code	nvarchar (50)	进货单编码

供应退货表（Prdt>Returns\_To\_Supplier）

字段	类型	说明
bar_code	nvarchar (50)	商品条形码
region	nvarchar (50)	区域
cnt	nvarchar (50)	商品数量
supplier_code	nvarchar (50)	供应商编码
unit_price	decimal (9, 2)	商品单价
date	nvarchar (50)	退货日期

operator	nvarchar (50)	经办人
order_code	nvarchar (50)	退货单编码

**销售表 (Prdt\_Sale\_Recods)**

字段	类型	说明
bar_code	nvarchar (50)	商品条形码
sale_cnt	Int	销售数量
unit_price	decimal (9, 2)	销售价格
date	nvarchar (50)	销售日期
region	nvarchar (50)	区域
pos_code	nvarchar (50)	POS 机编码
operator	nvarchar (50)	经办人
order_code	nvarchar (50)	售货单编码

**销售退货表 (Prdt\_Sale>Returns)**

字段	类型	说明
bar_code	nvarchar (50)	商品条形码
sale_cnt	Int	退货数量
unit_price	decimal (9, 2)	退货价格
date	nvarchar (50)	退货日期
region	nvarchar (50)	区域
pos_code	nvarchar (50)	POS 机编码
operator	nvarchar (50)	经办人
order_code	nvarchar (50)	退货单编码

**POS 机信息表 (POS\_Infomation)**

字段	类型	说明
pos_code	nvarchar (50)	POS 机编码
region	nvarchar (50)	区域
pos_name	nvarchar (50)	POS 机名称

<b>operator</b>	nvarchar (50)	操作员
-----------------	---------------	-----

**工作人员表 (Staffs\_Infomation)**

字段	类型	说明
<b>user_id</b>	nvarchar (50)	用户 ID
<b>user_name</b>	nvarchar (50)	用户姓名
<b>password</b>	nvarchar (50)	登录密码
<b>user_privi</b>	nvarchar (50)	用户权限
<b>id_number</b>	nvarchar (50)	身份证号
<b>addr</b>	nvarchar (50)	地址
<b>region</b>	nvarchar (50)	区域
<b>phone_num</b>	nvarchar (50)	联系电话

**数据库表 (Database\_Infomation)**

字段	类型	说明
<b>table_name</b>	nvarchar (50)	表明
<b>slices</b>	nvarchar (50)	分片
<b>redundancy</b>	nvarchar (50)	冗余

### 3.5、 分片与位置分配设计

根据以上得到的关于主要信息数据的需求分析结果，为该连锁超市系统的分布式数据库系统进行主要信息数据的分片和分配设计如下：

#### 1) 数据的分片设计

- A) 由于该连锁超市系统的各连锁店之间在经营上是独立的，每个门店只关心自己的经营状况，有关供应商信息、POS 机信息、商品信息和进货/销货信息等基础数据都是各门店单独管理和使用，门店之间互不相关。因此，商品表、供应商表、POS 机表、库存表、进货/销售开单表、进货/销售商品表、进货/销售退货商品表按照地域（门店所在区域标志）采用水平分片的方法得到水平片段。
- B) 这里我们对于商品信息进行了垂直分片，分成了商品表和库存表，因为商品的某些属性，例如库存量等需要经常更新，因此将这些属性划分出来构成单独的实体



可以减少系统开销。

- C) 由于整个连锁超市的职员信息由总公司管理和维护,各门店只可以查询本店的职员信息。所以职员信息不必分片,可以采用视图的形式提供给各门店查询本门店的职员信息。另外,各个区域数据库服务器的 IP 地址信息也只是由总部管理、维护,所以也不必分片。
- D) 由于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都要经常使用。因此,商品类别信息数据也不必分片。
- E) 该分布式数据库系统实现了简单的目录管理,记录数据库中各个表的分片和分配信息,以便数据更新时,维护各个站点上数据的一致性。这个表由总部规划建立,各门店也会经常使用。因此,数据库目录信息表也不必分片。

## 2) 数据及其片段的分配设计

- A) 对于只在各门店单独使用的除商品类别信息和数据库目录信息以外的其他基础信息的片段,采用按区域分片然后分配到各个门店的数据库服务器上。总站上有所有门店的所有信息。
- B) 整个系统的职员信息、IP 地址信息由公司总部管理和维护,所以只分配在总部站点中。
- C) 商品类别信息、数据库目录信息是由总部统一规定并下发到各门店的,由于各店经常会使用到这类基础信息,因此在各门店都具有相同的副本。所以,商品类别信息、数据库目录信息都不会分片但被复制,且复制的个数为门店的个数。

## 4、 分布式数据库查询优化策略；

### 4.1、 基本概念

分布式查询技术主要把用户提交的全局查询请求翻译为几个相关节点都可以识别的本地查询请求，以及把各个节点的查询结果汇总返回的问题，它包括分布式查询处理和分布式查询优化。分布式查询处理研究整个分布式查询处理的过程和策略；分布式查询优化研究查询策略的优化问题，就是要保证查询的总开销和总时间为最小。查询优化的主要任务是控制和加快查询的执行和数据的传输过程。

然而，查询优化一直是个复杂的问题，理想的全面的查询优化几乎是不可能的，往往只能达到局部目标的查询优化效果。许多专家和学者在这一领域曾做出过不少的研究和探讨。

查询优化的基本类型通常包括两类：

- **针对查询执行代价进行优化**，是使查询执行所使用的系统资源(总和)尽量地少，从而降低系统开销，整个系统的开销可以从单个系统资源的开销表达式中推出。
- **针对查询响应时间优化**，是尽量减少查询的响应时间，而不计较系统资源的耗费。

在分布式数据库系统中，一方面，许多相对独立的处理器可能参与数据库操作。由于在处理一个问题时可以使用多台机器，并行以及加快查询反应速度的可能性增大。由于数据可以在多个节点上存在副本，系统可能不会仅仅由于一个节点或部件发生故障而不得不停止处理。

另一方面，分布式处理增加了分布式系统各个方面的复杂性，因此即使是 DBMS 中最基本的组成部分的设计，也得重新考虑。在许多分布式环境中，通信开销可能远大于处理开销，因此消息如何传送是一个基本的研究课题，比如分布式提交和分布式封锁。

### 4.2、 用于连锁超市的分布式数据库的查询优化

#### 4.2.1、 基于关系代数等价变换的优化算法

这是一种针对查询代价进行优化的一种常用算法。基于关系代数等价变换的优化算法是一种既能减少操作量又能减少操作次数的算法。基于关系代数等价变换优化算法的基本原理把查询问题转变为关系代数表达式，分析得到查询树（语法树），进行从全局到片段的变换得到基于片段上的查询树，然后利用关系代数等价变换规则的优化算法，尽可能先执行选择和

投影操作。这样，一方面可以减少其后操作的操作量，另一方面可以减少操作次数。对该查询树进行优化。从而达到查询优化的目的。

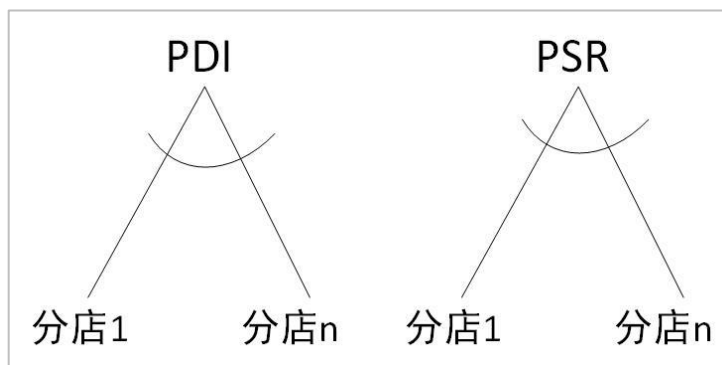
例如在基于连锁超市的分布式数据库中查询某一商品（prdt\_name=“香脆锅巴”）在 2018 年 1 月的销售量。该信息可有每个门店的销售表查询得到，这个查询问题的 SQL 语句如下：

```
SELECT SUM ( PSR.sale_cnt ) from Prdt_Detail_Infomation as PDI,
Prdt_Sale_Recods as PSR where PDI.prdt_id = PSR.prdt_id AND PDI.prdt_name
= “香脆锅巴” AND PSR.sale_date=“2018.01”;
```

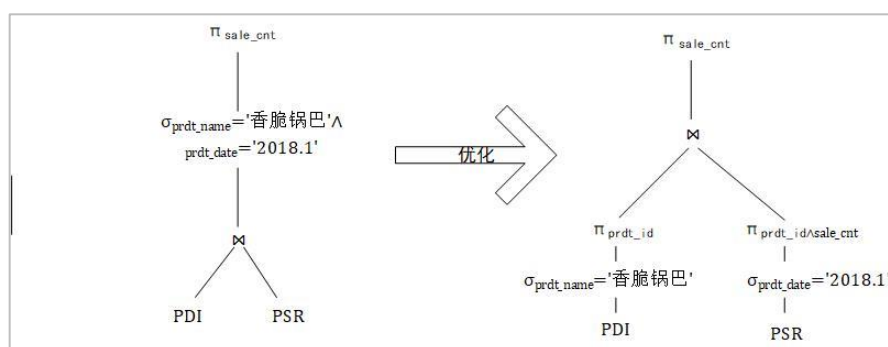
它的代数关系运算如下：

$$\pi_{\text{sale\_cnt}}(\sigma_{\text{prdt\_name}='香脆锅巴' \wedge \text{prdt\_date}='2018.1'}(\sigma_{\text{PDI.prdt\_id}\#=\text{PSR.prdt\_id}\#}(\text{PDI} \bowtie \text{PSR})))$$

由数据库设计可以知道商品表和销售表都是采用水平分片的。如下图所示：



代数运算关系中的乘积操作合并成了一个自然连接操作，转换成相应片段上的查询树如下图所示左图所示。根据基于关系代数等价变换的优化算法进行优化，优化后的查询如下图所示右图所示。



关系代数等价变换规则的优化算法就是利用关系代数等价变换规则，把查询树中连接和合并操作尽可能上提（向树根方向移）。选择和投影操作尽可能下移（向树叶方向移）到片段的定义处。这就是说，尽可能先执行选择和投影操作，后执行连接和合并操作。经过选择和投影操作不但可以减少其后操作的操作量，而且还可以减少操作次数。

#### 4.2.2、 基于半连接操作的查询优化算法

这是一种针对查询响应时间进行优化的一种常用算法。基于半连接操作的查询优化的思想是经过半连接操作，可减少操作关系的数据量，从而减少站点间数据的传输量。

数据在整个关系在网络中传输，这显然是一种冗余的方法，在一个关系传输到另一场地后，并非每个数据都参与连接操作或都是有用，因此不参与连接的数据或无用的数据不必在网络中来回传输。基于半连接的优化策略的原理就是采用半连接操作，在网络中只传输参与连接的数据。连接查询的优化问题几乎是分布式数据库的分布式查询优化算法的全部，在分布式数据库中连接查询的主要手段是半连接技术，各种不同算法的差异主要是在连接顺序上，即在保证结果一致的情况下，以什么样的顺序将这些表连接起来最优。优化的对象一般是数据传输量的总和。

另外当在两个关系 **R** 和 **S** 上实施某个操作例如连接时，有多个可选方案而且必须选择其中之一时，一些可能的选项如下：

- A) 可以将 **S** 复制到 **R** 所在节点，并在该节点执行计算。
- B) 可以将 **R** 复制到 **S** 所在节点，并在该节点执行计算。
- C) 可以将 **R** 和 **S** 复制到二者各自所在节点之外的第三个节点，并在该节点执行计算。

哪种选择最好，这依赖于多个因素，其中包括哪个节点上有可用的处理时间以及操作结果是否需要与第三个结点上的数据相结合等。

例如在本基于连锁超市的分布式数据库中，为了减少在网络中传输的数据量，一方面尽可能的使用半连接从简减少不必要的数据传输花费。对上述查询某一商品（**prdt\_name**=“香脆锅巴”）在 2018 年 1 月的销售量。该信息可有每个门店的销售表查询得到，这个查询问题的 SQL 语句写为如下：

```
SELECT SUM (PS.sale_cnt) from
  (SELECT PSR.prdt_Id, PSR.sale_cnt FROM Prdt_Sale_Recods as PSR WHERE
  PSR.sale_date=" 2018.01" ) AS PS WHERE PS.prdt_id IN
  (SELECT prdt_id FROM Prdt_Detail_Infomation as PDI WHERE PDI.prdt_name
  = "香脆锅巴")
```

另外对于某些商品在个别分店的销量特别大，那么对于该类商品的查询计算应当放置在该连锁分店，计算结束后将结果返回。这样也可极大地减少数据在网络中的花费。相比传输大量的中间数据，直接传输结果数据则显得更加有效。

## 5、 事务处理方式；

### 5.1、 基本概念

分布式数据库系统是由若干相互自治的单节点数据库系统松散耦合成的一个总的数据库系统。其中数据以数据片段/复制的形式分布于各个节点上。事务是访问并可能更新各种数据项的一个程序执行单元。分布式事务具有访问共享数据的能力，即它可以访问存储于远程节点上的数据。

事务可分两种：

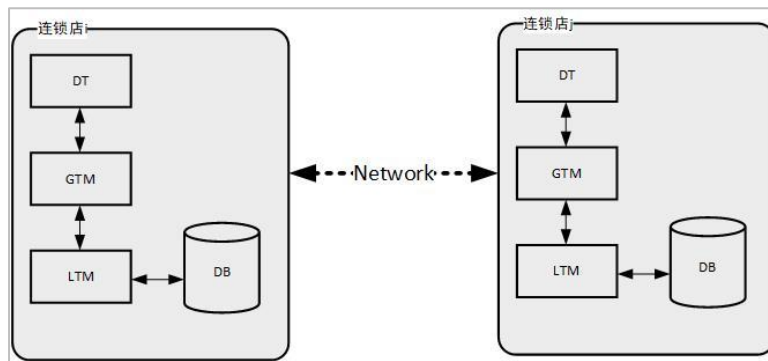
- **局部事务：** 仅仅访问和更新一个节点中数据的事务叫；
- **全局事务：** 需要访问和更新多个节点中数据的事务称；

分布式数据库研究的主要问题之一 是事务处理,即事务的调度必须同时满足事务的原子性、一致性、隔离性和持久性(简称为 **ACID** 特性)。各个分布的节点具有局部自治性，即能够管理局部事务，并参与全局事务的执行，保证数据的全局一致性。

### 5.2、 分布式事务处理流程

在本应于与连锁超市的分布式数据库中，各个节点安装的局部的数据库管理软件 **SQL Server**。虽然总的分布式数据库系统集成了一组自治的数据库系统来提供全局的数据库功能，在 **DDBS** 环境下，事务管理分为全局和局部两层。但是作为已存在的局部数据库之一，任何一个 **DDBS** 最先关心的是保护局部数据库的自治性。所以这就通过在局部数据库集 (**LDBSs**) 上叠加全局事务管理(**GTM**)来实现的。

全局事务提交给全局事务管理,在那里它们被分解为全局事务的子事务被个别地提交给局部节点的局部事务管理系统。在同 一时间，各个局部事务被提交给局部事务管理系统。每个局部事务管理系统保证在它的节点正确执行局部子事务，全局事务管理保证全局事务的正确执行。



本应于与连锁超市的分布式数据库系统的模式上图所示，最底层是本地事务管理程序 (LTM)，中间一层是全局事务管理程序 (GTM)，最高层是分布式事务 (DT)。

为实现分布式事务的原子性，每个连锁店的数据库系统中建立一个能实现本地事务的本地事务管理程序 (LTM)，建立分布式事务管理程序来实现开始事务、提交和取消等原语，确保子事务的原子性。

其原语表示如下：

- **开始事务：**当根代理者发出一事务时，GTM 就必须发出一局部开始原语给原发节点的 LTM 和其他活跃的连锁店节点，这样把全部代理者变换为子事务；从此时起，位于同一分布式事务被激活任一新的连锁店代理者时，都要求把局部开始发给要该连锁店节点的 LTM，所以该新代理者就创建为一子事务。
- **取消：**当某一连锁店发出中止时，所有现有的子事务都必须中止。这是通过向所有活跃的连锁店节点的 LTM 发送局部取消来实现的。这样通过全部子事务获得了一全局中止的效果。因为这些代理者自开始事务起所执行的全部动作都被 LTM 取消了。
- **提交：**分布式事务的正确提交要求它的子事务甚至在故障情况下也能本地提交。所以不允许一个子事务因为故障而本地中止，其它的子事务却都被提交。

## 6、 并发控制处理机制

### 6.1、 基本概念

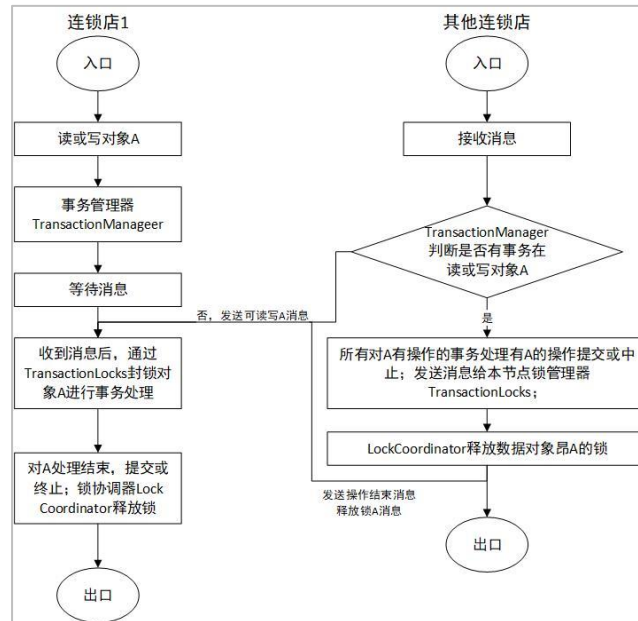
本应用于连锁超市的分布式数据库是一个共享的资源，允许多个用户程序并行的操作各个连锁分店节点的数据库。这就会产生多个事务并发地存取同一数据的情况，若对并发操作不加控制，就会破坏结果的正确性和数据库的完整性。但如果不并发操作，则数据库资源使用效率太低。为保证事务的正确提交，分布式数据库的每个节点必须采用并发控制协议。

分布式 DBMS 的分布式并发控制机制是确保数据库的一致性和并发的高效性。串行性是判定并发控制正确性的原则，当多个事务并发执行的结果与按某一次序串行执行其结果相同，则认为并发操作是正确的，并称这种调度策略为可串行化调度。在本应用于连锁超市的分布式数据库系统中，对并发操作采取的措施为两阶段封锁机制(2PL)。

### 6.2、 并发控制方法

#### 6.2.1、 两阶段封锁机制(2PL)

在构建的本应用于连锁超市的分布式数据库系统中，我们采用分布式两段封锁协议(D2PL)进行并发控制，从而对每个节点的锁管理器进行有效调度，保证并发事务的可串行性，全局数据的一致性。设连锁分店1中应用程序要读或写目标数据对象A的某些属性，而对象A的部分属性可能正被其它参与节点中的事务进行读或写操作。其分布式节点之间处理并发事务的控制流程结构如下图所示。



分布式事务协调器向每一个参与节点的锁管理器发送“封锁请求”消息，参与节点的数据处理器调度处理结束后向事务协调器发送“操作结束”消息，同时发送到自己的锁管理器，然后释放锁并通知事务协调器，协调并发事务的执行。

### 6.2.2、 其他并发控制协议

例如时间戳协议，即每个节点用系统时钟值或者逻辑计数器产生唯一的局部时间戳,每个事务被分配一个时间表,事务的时间戳就决定了串行化顺序。还有乐观并发控制算法等等。



## 7、 分布式数据库系统的安全性、可靠性

本分布式数据库系统主要起着为连锁超市提供基本的信息和数据服务的作用，并且能实现数据共享。本系统是由服务器和若干个客户机组成，并通过网络互连。

### 7.1、 安全性

本分布式数据库应用于连锁超市管理系统时，超市的各个位于不同区域，系统中的一台客户机互连网访问其他数据时，不仅需要对在网上传输的数据进行加密，而且要经过数据库系统的确认，才能有效的进行访问从而具有良好的安全性。对于安全性的控制在本系统中主要包含以下几点。

- 站点、服务器之间的身份验证
- 通讯双方之间采用对称密码算法建立保密信道
- 用户访问控制
- 库文数据加密

### 7.2、 可靠性

分布式数据库的组织方法有很多优点，它不但降低了数据传输的代价，而且提高了数据库系统的稳定性，并且易于维护。另外如果需要增加新的数据库或者扩充新的小型计算机，利用分布式数据库系统都比较容易实现。

首先不同级别的连锁分店可以对其自身数据实行局部控制。例如：不同地方的超市连锁分店在本地录入、查询、维护时，由于计算机资源靠近用户，可以降低通信代价，提高响应速度，而涉及其他场地数据库中的数据只是少量的，从而可以大大减少网络上的信息传输量。

其次系统经济，可靠性高，可用性好。与一个大型计算机支持一个大型数据库再加一些进程和远程终端的传统集中式数据库相比，由超级微型计算机或超级小型计算机支持的分布式数据库系统往往具有更高的性价比和实施灵活性。分布式系统比集中式系统具有更高的可靠性和更好的可用性。如由于数据分布在多个场地并有许多复制数据，在个别场地或个别通信链路发生故障时，不至于导致整个系统的崩溃，而且系统的局部故障不会引起全局失控。

最后在一定条件下响应速度加快。如果存取的数据在本地数据库中，那末就可以由用户所在的计算机来执行，速度就快。可扩展性好，易于集成现有系统，也易于扩充。