第2章　数据仓库环境

数据仓库是体系结构设计环境的核心，是决策支持系统 (DSS)处理的基础。因为在数据仓库中只有单一集成的数据资源，并且因为数据是可访问的，所以与传统数据环境相比，在数据仓库环境中DSS分析员的工作将要容易得多。

本章将叙述数据仓库问题的一些重要特性。

数据仓库是一个面向主题的、集成的、非易失的且随时间变化的数据集合，用来支持管理人员的决策。

数据仓库的面向主题性，如图 2-1所示。

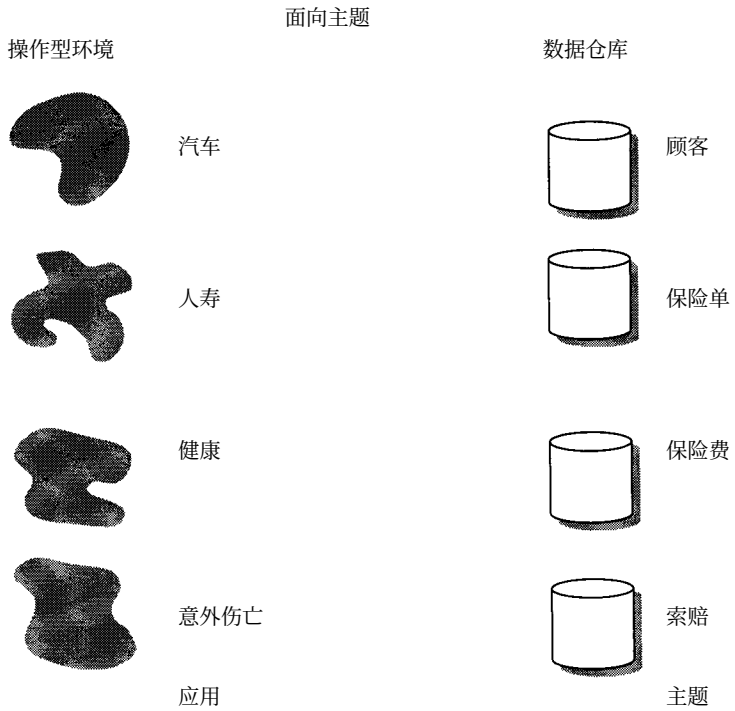


图2-1　数据面向主题的一个例子

传统的操作型系统是围绕公司的应用进行组织的。对一个保险公司来说，应用问题可能是汽车保险、健康保险、人寿保险与意外伤亡保险。公司的主要主题范围可能是顾客、保险单、保险费与索赔。

数据仓库的第二个显著特点是集成的。在数据仓库的所有特性之中，这是最重要的。图2-2说明了当数据由面向应用的操作型环境向数据仓库传送时所进行的集成。

应用问题的设计人员历经多年制定出来的不同的设计决策有很多很多种不同的表示方法，没有什么应用在编码、命名习惯、实际属性、属性度量等方面是一致的，各个应用问题设计

员自由地做出他或她自己的设计决策。

当数据进入数据仓库时，要采用某种方法来消除应用问题中的许多不一致性。例如，在图2-2中，考虑关于“性别”的编码，在数据仓库中是编码为 m/f还是1/0并不重要，重要的是，无论什么原始应用问题，无论数据仓库如何进行编码，在数据仓库中应该一致地进行编码。如果应用数据编码为X/Y，当其进入数据仓库时就要进行转换。对所有的应用设计问题都要考虑同样的一致性处理，比如命名习惯、键码结构、属性度量以及数据特点等。

数据仓库的第三个重要特性是数据仓库是非易失的。图 2-3说明了数据的非易失性。

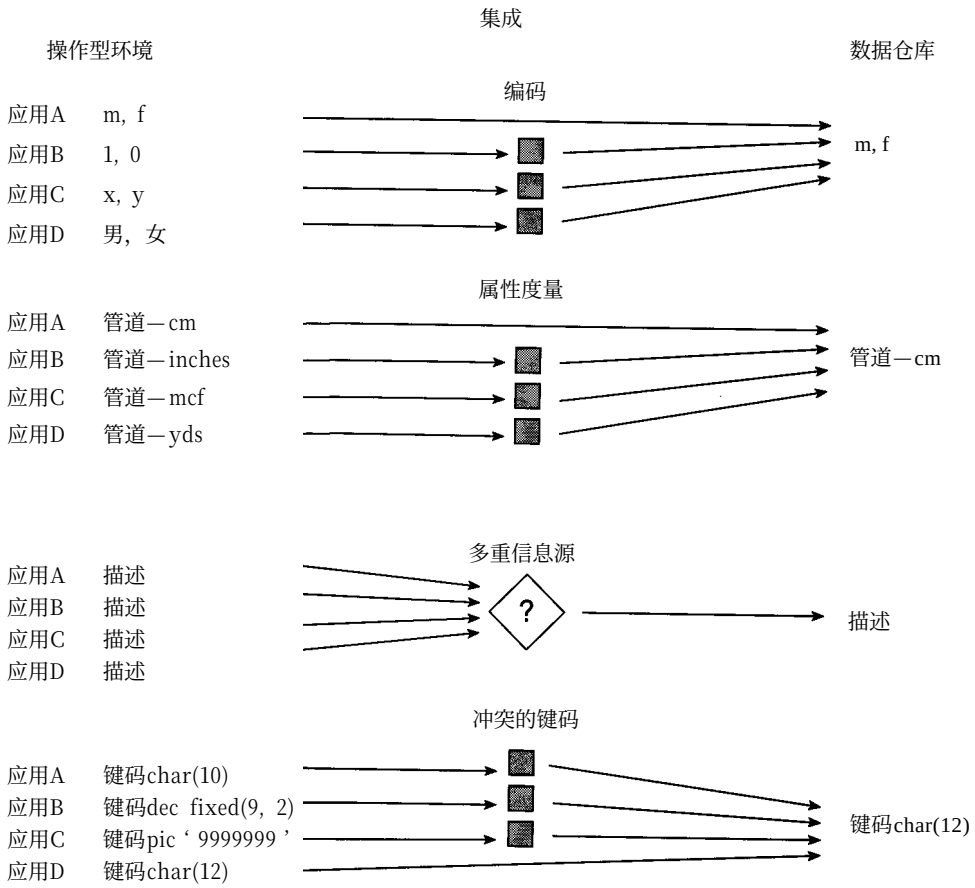


图2-2 集成问题

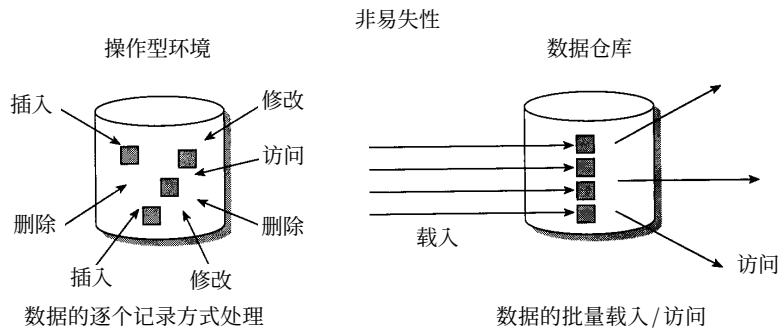


图2-3 非易失性问题

图2-3表示了操作型数据正规地是一次访问和处理一个记录。可以对操作型环境中的数据进行更新。但数据仓库中的数据呈现出非常不同的特性。数据仓库的数据通常是一起载入与访问的，但在数据仓库环境中并不进行一般意义上的数据更新。

数据仓库的最后一个显著特性是其随时间的变化性。如图 2-4所示。数据仓库中的数据随时间变化的特性表现在以下几个方面：

- 数据仓库中的数据时间期限要远远长于操作型系统中的数据时间期限。操作型系统的时间期限一般是60～90天，而数据仓库中数据的时间期限通常是 5～10年。
- 操作型数据库含有“ 当前值 ”的数据，这些数据的准确性在访问时是有效的，同样当前值的数据能被更新。而数据仓库中的数据仅仅是一系列某一时刻生成的复杂的快照。
- 操作型数据的键码结构可能包含也可能不包含时间元素，如年、月、日等。而数据仓库的键码结构总是包含某时间元素。

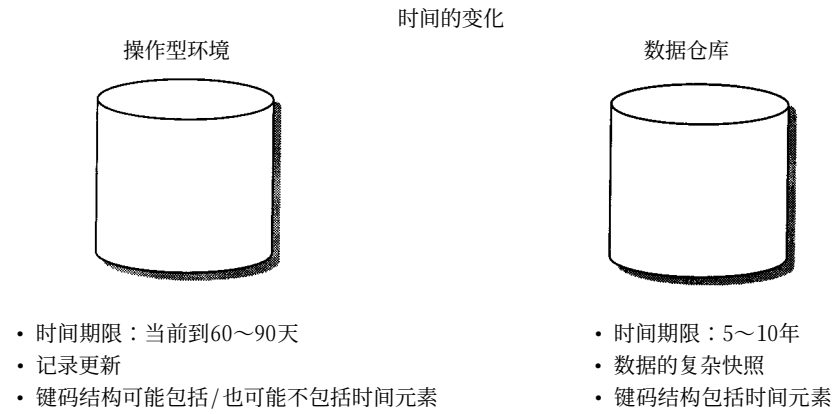


图2-4 随时间的变化问题

2.1 数据仓库的结构

数据仓库的结构如图2-5所示。图2-5表明，在数据仓库中数据存在着不同的细节级：早期

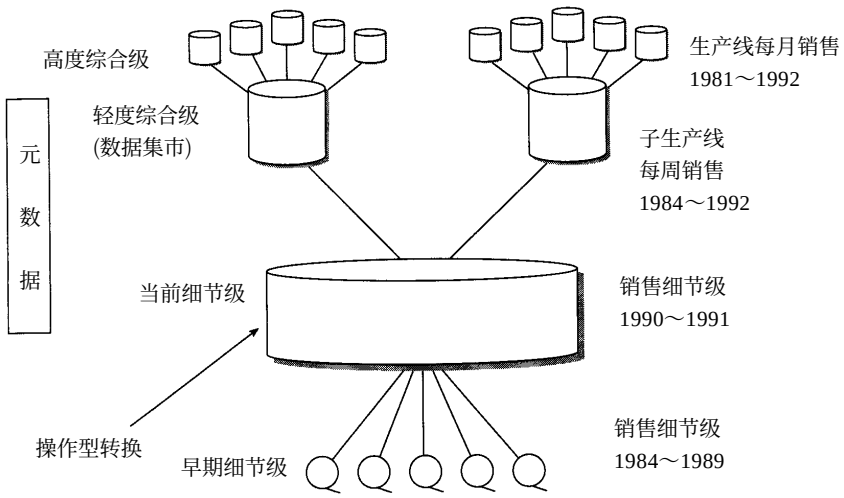


图2-5 数据仓库的结构

细节级(通常是备用的、批量的存储)、当前细节级、轻度综合数据级(数据集市)以及高度综合数据级。数据是由操作型环境导入数据仓库的。相当数量的数据转换通常发生在由操作型级别向数据仓库级别传输过程中。

一旦数据过期，就由当前细节级进入早期细节级。综合后的数据由当前细节级进入轻度综合数据级，然后由轻度综合数据级进入高度综合数据级。

2.2 面向主题

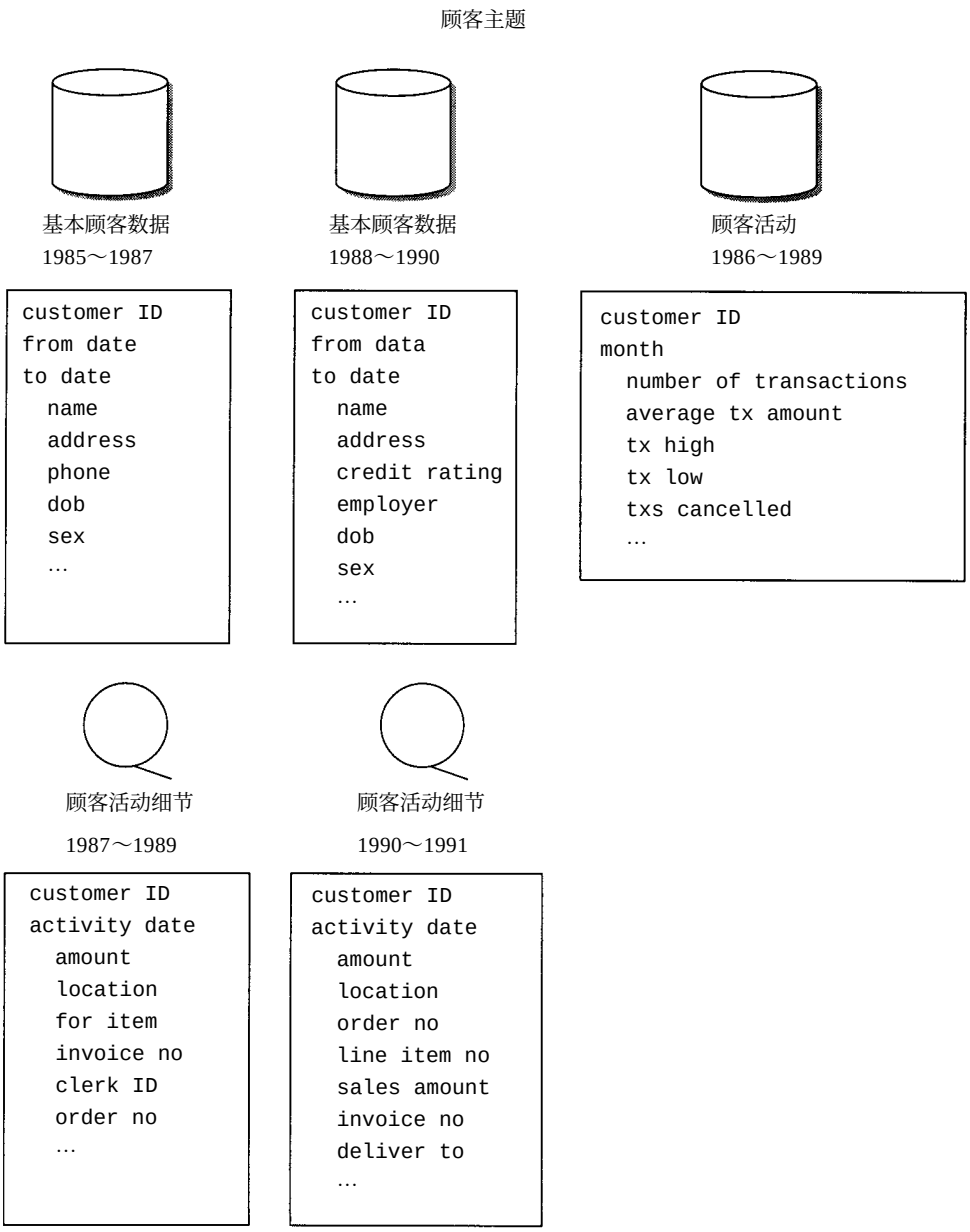


图2-6 数据仓库数据用主要主题领域(这里用顾客)来组织

数据仓库是面向在数据模型中已定义好的公司的主要主题领域的。典型的主题领域包括：

- 顾客。
- 产品。
- 事务或活动。
- 保险单。
- 索赔。
- 帐目。

在数据仓库中，主要主题领域是以一组相关的表来具体实现的。例如，一个顾客主题的实现可能像图2-6所示。

在图2-6中有五个相关的表，每个表设计用来实现作为顾客这个主要主题领域的一部分。有一张从1985年～1987年定义的顾客信息基本表，另有一张 1988年～1990年之间的顾客数据定义基本表。有一张 1986年～1989年间的累积的顾客活动表。根据顾客每月的活动，针对每个顾客记录编制一张月度总结记录表。

这里我们有从1987年～1989年和从1990年～1991年的顾客的详细活动文件。文件中数据的定义因年份不同而不同。

一个顾客的所有表通过一个公共键码联系起来，图 2-7表明了用公共键码顾客标识号

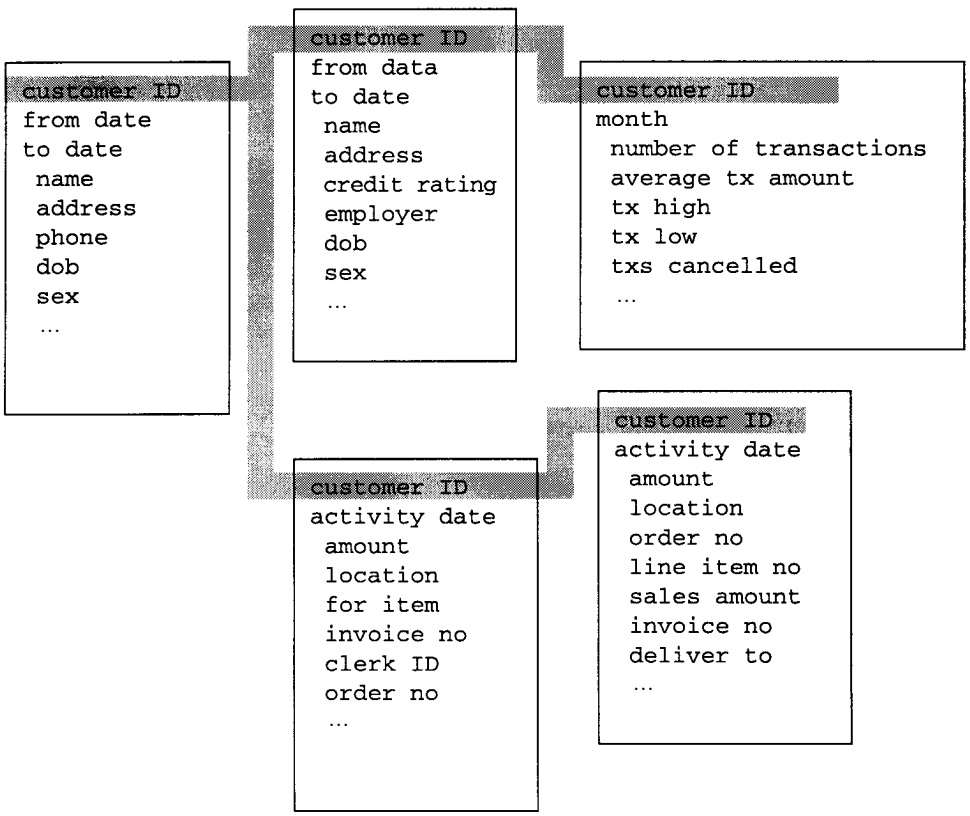


图2-7 属于相同主题域的数据集合由一个公共键码联系起来

(customer ID)将在顾客主题领域中所找到的所有数据联系起来。顾客主题领域另一个有趣的特征是其数据可以存储在不同的介质上，如图 2-8所示。

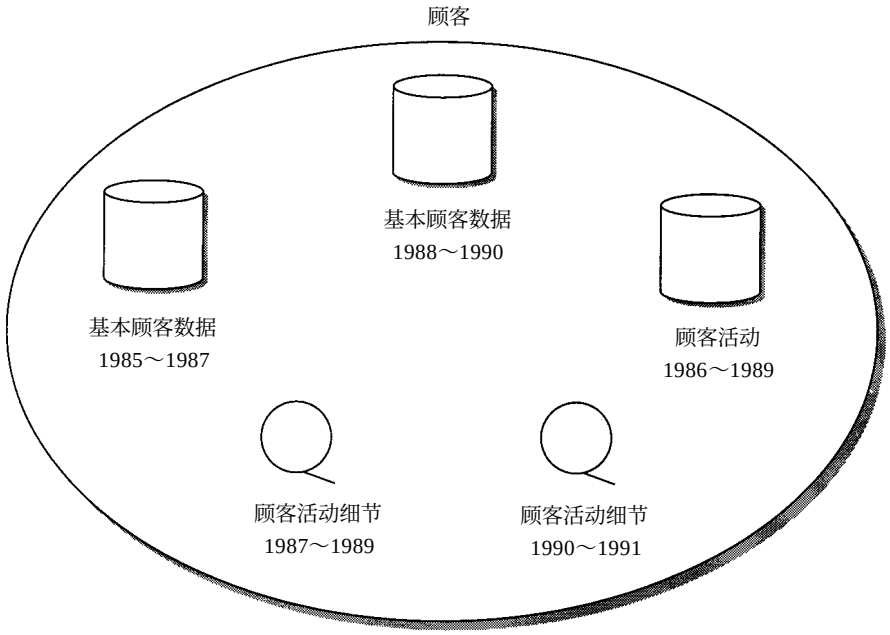


图2-8 数据仓库中主题领域可能包含不同介质上的数据

图2-8表明有的数据存储在直接存取存储设备 (DASD)上，有的数据存储在磁带上。数据存储在不同介质上意味着在数据仓库中可能有多个数据库管理系统 (DBMS)对数据进行管理，或者某些数据根本没有被某个 DBMS管理。不能仅仅因为数据存储在磁带上，就认为它不是数据仓库的一部分。

访问概率高且存储空间小的数据存放在快速且相对昂贵的存储介质上；访问概率低且存储空间大的数据存放在廉价、慢速的存储介质上。

通常，DASD和磁带是数据仓库中两种最常见的数据存储介质，但并非只能用这些介质，另外两个不可忽视的介质是缩微胶片和光盘。缩微胶片适于存储详细的且无需在电子媒体中再次复制的记录，合法的记录经常在缩微胶片上存放一个不确定的时期。光盘特别适合于用作数据仓库的存储介质，因为它价廉、速度较快且能存储大量的数据。另外因为数据仓库中的数据一旦载入几乎从来不用更新，这个特征使光盘存储成为数据仓库非常理想的选择。

这些文件(如图2-8所示)另一个有趣的特征是相同的数据既有综合级，又有细节级。每月活动是综合的。同时，支持每月活动的细节存放在数据的磁带级上。这就是后面要讨论的“粒度转换”的一种形式。

当数据围绕顾客这个主题组织时，每个键码都有一个时间元素。如图 2-9所示。

一些表是在“从日期到日期”的基础上进行组织的，称之为数据的连续组织。另一些表是在“每月累积”的基础上进行组织的。还有一些是在“记录或活动的单独日期”的基础上组织的。但是，所有记录都有某种形式的日期连接到键码，通常是键码的较低部分。

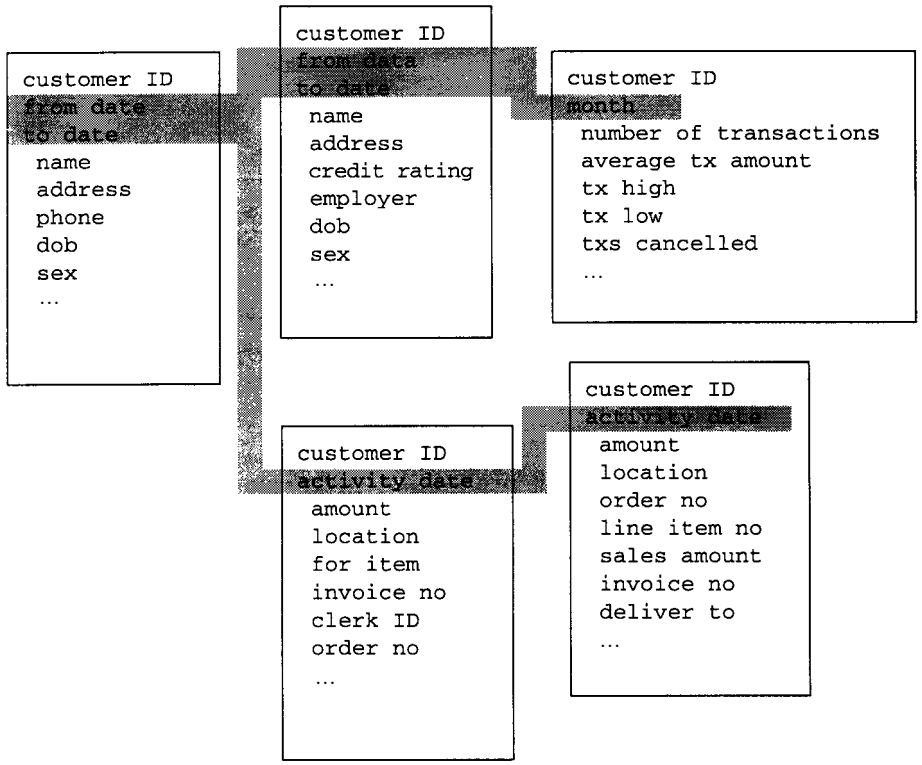


图2-9 数据仓库中每个表都有时间元素作为键码结构的一部分(通常为较低的部分)

2.3 第1天到第n天的现象

建立数据仓库不是一蹴而就的。相反，数据仓库只能一次一步地进行设计和载入数据，即它是进化性的，而非革命性的。突然建立一个数据仓库的费用、需要的资源和对环境的破坏，都表明数据仓库的建立要采用有序地反复和一次一步的方式。

图2-10说明一个建立数据仓库的典型过程。第 1天，通晓本质上进行操作型处理的几个系统。第2天，对数据仓库中第一个主题领域的最初几个表载入数据，此时就会产生一定的好奇心，用户开始发现数据仓库和分析处理。

第3天，更多的数据载入数据仓库，并且随着数据量增大，将吸引更多的用户。一旦用户发现有较容易载入的集成数据源，并有在时间维上观察数据的历史基础，这就不仅仅是好奇心了。大约此时，认真的DSS分析员渐渐地被吸引到数据仓库中。

第4天，随着更多的数据载入数据仓库，一批存储在操作型环境的数据被适当地放入数据仓库中。现在，我们就“发现”数据仓库是可用来进行分析处理的信息源。各种各样的 DSS应用出现了。的确，伴随着现在存入数据仓库的大规模数据，此时开始出现如此多的用户和如此多的处理请求，以致于一些用户进入数据仓库的要求和分析工作被推迟。进入数据仓库的竞争成为使用数据仓库的障碍。

第5天，部门数据库 (数据集市，或OLAP)开始兴起，各部门发现通过把数据从数据仓库输入它们自己的部门处理环境，会使它们的处理既便宜又容易。到达部门级的数据吸引着一

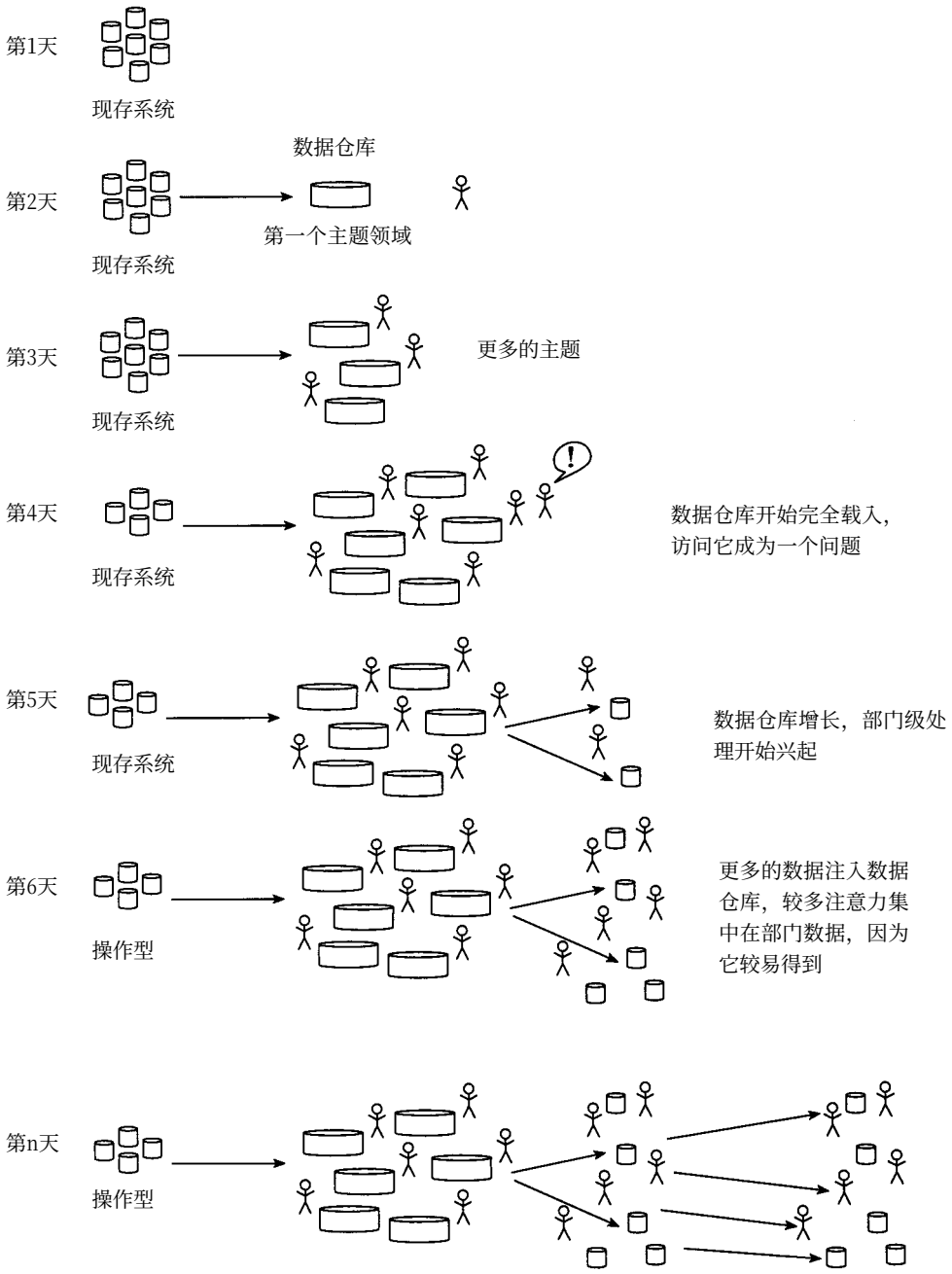


图2-10 第1天到第n天的现象

些DSS分析员。

第6天，部门系统出现繁忙，得到部门数据比获得数据仓库的数据更便宜、更快、更容易。很快最终用户就放弃数据仓库的细节，去进行部门处理。

第n天，这种体系结构得到充分发展。生产系统的原始集合中只剩下操作型处理。数据仓库具有丰富的数据，并有一些数据仓库的直接用户和许多部门数据库。因为在部门级上获得

处理所需要的数据既容易又便宜，所以大部分 DSS分析处理都在部门级进行。

当然，从第1天到第n天的进化需要很长的时间，通常需要几年。并且在从第 1天到第n天的处理过程中，DSS环境在不断地提高和职能化。

(正常情况下，此时会发现蜘蛛网开始发展成为一个更大、更壮观的形式。但情况并非完全如此，尽管解释起来相当复杂。要了解进一步深入的解释，请参考“ The cabinet effcct”, Data Base Programming Design,May 1991。)

2.4 粒度

粒度问题是设计数据仓库的一个最重要方面。粒度是指数据仓库的数据单位中保存数据的细化或综合程度的级别。细化程度越高，粒度级就越小；相反，细化程度越低，粒度级就越大。

数据的粒度一直是一个设计问题。在早期建立的操作型系统中，粒度是用于访问授权的。当详细的数据被更新时，几乎总是把它存放在最低粒度级上。但在数据仓库环境中，对粒度不作假设。图2-11说明了粒度问题。

在数据仓库环境中粒度之所以是主要的设计问题，是因为它深深地影响存放在数据仓库中的数据量的大小，同时影响数据仓库所能回答的查询类型。在数据仓库中的数据量大小与查询的详细程度之间要作出权衡。

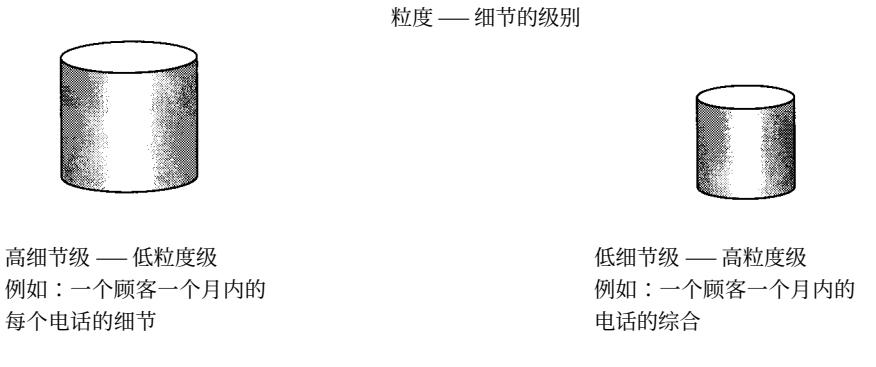


图2-11 数据仓库主要设计问题: 粒度、分割和适当设计

2.4.1 粒度的一个例子

图2-12表示了粒度问题的一个例子。左边是一个低粒度级，每个活动 (在这里是一次电话) 被详细记录下来，数据的格式如图所示。到月底每个顾客平均有 200条记录(全月中每个电话都记录一次)，因而总共需要40 000个字节。

该图的右边是一个高粒度级。数据代表一位顾客一个月的综合信息，每位顾客一个月只有一个记录，这样的记录大约只需200个字节，记录的格式如图所示。

显然，如果数据仓库的空间很有限的话 (数据量总是数据仓库中的首要问题)，用高粒度级表示数据将比用低粒度级表示数据的效率要高得多。

高粒度级不仅只需要少得多的字节存放数据，而且只需要较少的索引项。然而数据量大小和原始空间问题不是仅有的应考虑的问题。为了访问大量数据，其处理能力的大小同样也是应考虑的一个因素。

所以，在数据仓库中数据压缩非常有用。当数据被压缩后就会大大节省所用的 DASD存储空间，节省所需的索引项，以及节省处理数据的处理器资源。

但是，当提高粒度级时，数据压缩就会出现另一个问题，图 2-13表示作出的选择。

在图2-13中,当提高数据粒度级时,数据所能回答查询的能力就会随之降低。换句话说,在一个很低的粒度级上你实际可以回答任何问题,但在高粒度级上,数据所能处理的问题的数量是有限的。

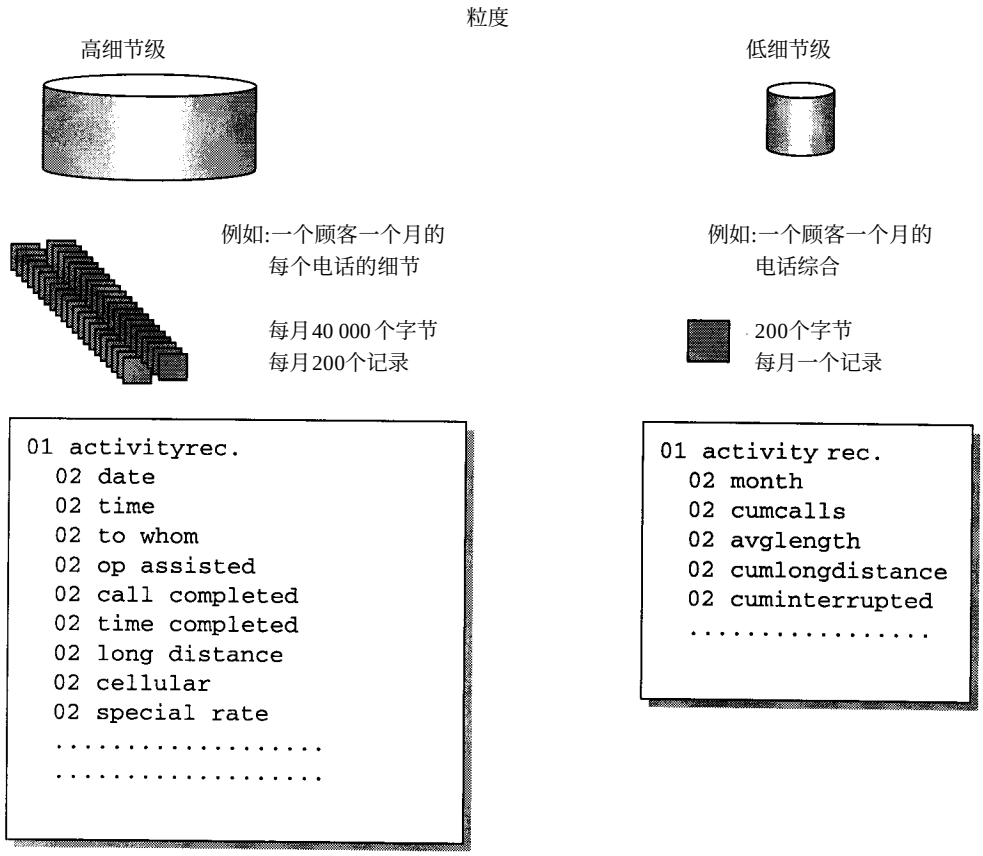


图2-12 确定粒度级是数据仓库环境中最重要的设计问题

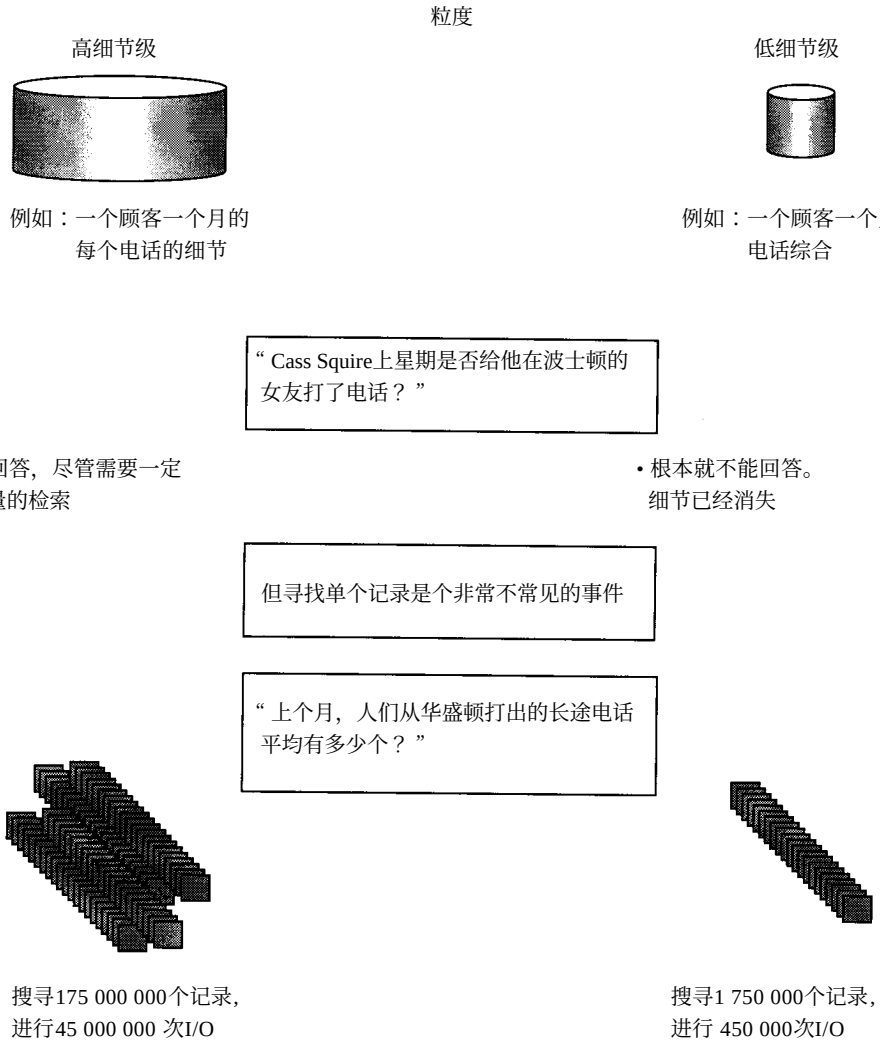


图2-13 粒度级对能回答什么问题和回答问题所需什么资源有深刻的影响

在图2-13中提出了下面的问题：

“ Cass Squire上星期是否给他在波士顿的女友打了电话？ ”

在低粒度级上，这个问题是可以回答的，虽然这种回答将花费大量资源去查阅大量的记录，但是Cass上周是否给他在波士顿的女友打了电话最终总是可以确定的。

然而，在高粒度级上就无法明确地回答这个问题。假如在数据仓库中存放的只是 Cass Squire打的电话总数，那么就不能确定其中是否有一个电话是打往波士顿的。

但是，在进行DSS处理时(这在数据仓库环境中是常见的)，很少对单个事件进行检查。通常是针对某种数据集合进行处理的，这意味着要查阅大量记录。

例如，假设提出下面的集合性查询问题：

“ 上个月人们从华盛顿打出的长途电话平均多少个？ ”

在一个DSS环境中这种查询类型是非常常见的。当然，它既可以在高粒度级上也可以在低粒度级上得到回答。但在回答这个问题时，在不同的粒度级上所使用的资源具有相当大的差

异。在低粒度级上回答这个问题需要查询每一个记录，所以需要大量的资源来回答这个问题。

但在高粒度级上，数据进行了很大的压缩，而且能够提供一个答案。如果在高粒度级上包括了足够的细节，则使用高粒度级数据的效率将会高得多。

在管理数据的粒度问题中的权衡如图 2-14所示。在设计和构造数据仓库之初就必须仔细考虑这种权衡。

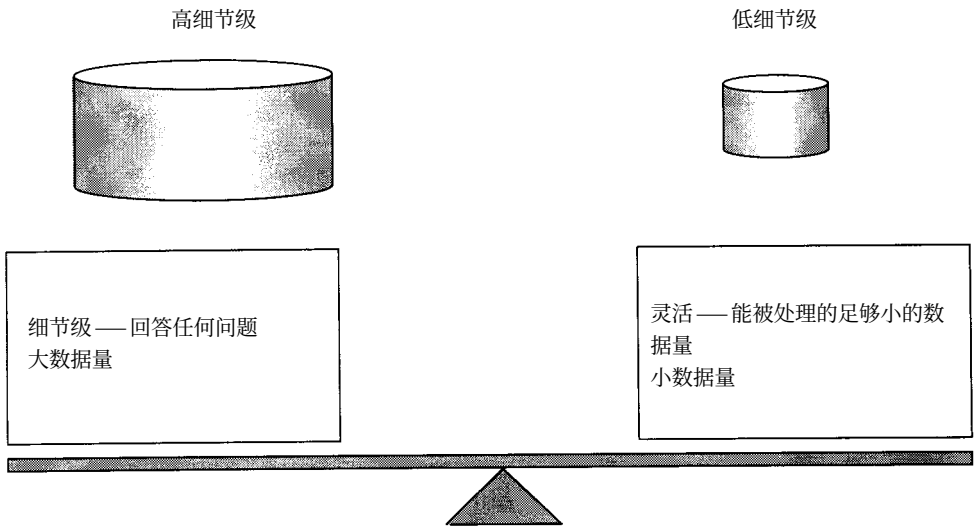


图2-14 粒度的权衡是首要的，所以大多数组织的最佳解决办法是采用多重粒度级的形式

2.4.2 粒度的双重级别

很多时候，十分需要提高存储与访问数据的效率，以及非常详细地分析数据的能力。 (换句话说，机构想有自己的蛋糕，并且吃了它！)当一个企业或组织的数据仓库中拥有大量数据时，在数据仓库的细节部分考虑双重 (或多重)粒度级是很有意义的。事实上，需要多个粒度级而不是一个粒度级的需求，是因为粒度级设计采用双重级别应该是几乎每个机构默认的选择。图2-15表明了在数据仓库的细节级上的两种粒度级。

图2-15所示(一家电话公司)的我们称之为“ 双重 ”粒度级的设计，能满足大多数机构的需要。在操作层是大量的细节，其中大部分细节是为了满足结帐系统的需求。多达 30多天的细节存放在这种操作层中。

在这个例中的数据仓库包括两种类型的数据：轻度综合数据和“ 真实档案 ”细节数据。数据仓库中的数据能回溯十年。从数据仓库中提取的数据是流向电话公司不同地区的“ 地区 ”数据，然后各地区独立地分析各自的数据。在个体级上进行各自的启发式分析处理。

什么是轻度综合级，现在可能还不是很明显。图 2-16表明了一种轻度综合级。

当数据从操作型环境(存储30天的数据)载入时，它就被顾客综合成可能用于 DSS分析的数据域。J.Jones的记录显示她每月打电话的次数、每个电话的平均长度、长途电话的次数、接线员帮助呼叫的次数，等等。

在轻度综合数据库中的数据量比细节数据库中的数据量少得多。当然，在轻度综合级数

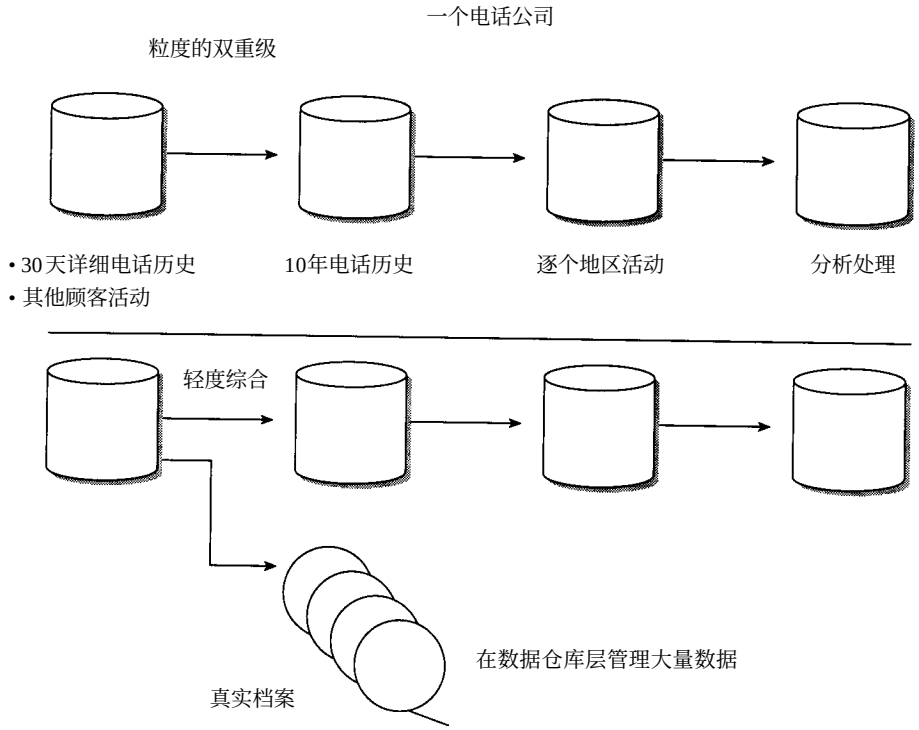
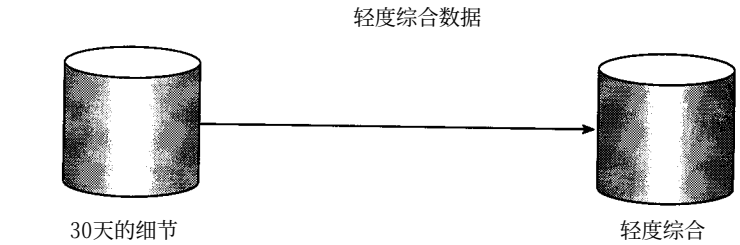


图2-15 大量数据使得大部分组织在数据仓库中需要两个粒度级



J Jones 4月12日 下午6:01～6:12 415-566-9982 接线员帮助 4月12日 下午6:15～6:16 415-334-8847 长途 4月12日 下午6:23～9:38 408-223-7745 4月13日 上午9:12～9:23 408-223-7745 4月13日 上午10:15～10:21 408-223-7745 接线员帮助	4月15日 上午11:01～11:21 415-964-4738 4月15日 上午11:39～12:01 703-570-5770 未接通 4月15日 下午12:10～12:46 703-841-5770 号码错误 4月16日 下午12:34～12:56 415-964-3130 ……	四月份 J Jones 电话数量——45个 电话的平均长度——14分钟 长途电话数——18个 接线员帮助呼叫数——2个 未接通电话数——1个

存储1个记录所需的字节数——225

每月一个顾客存储200个记录平均需要的字节数——45 000

图2-16 采用数据的轻度综合，使大量的数据可简洁地表示

据库中，对能访问的细节级存在一定的限制。

数据仓库中数据的第二层 —— 最低粒度级 —— 存放在数据的真实档案层上，如图 2-17所示。

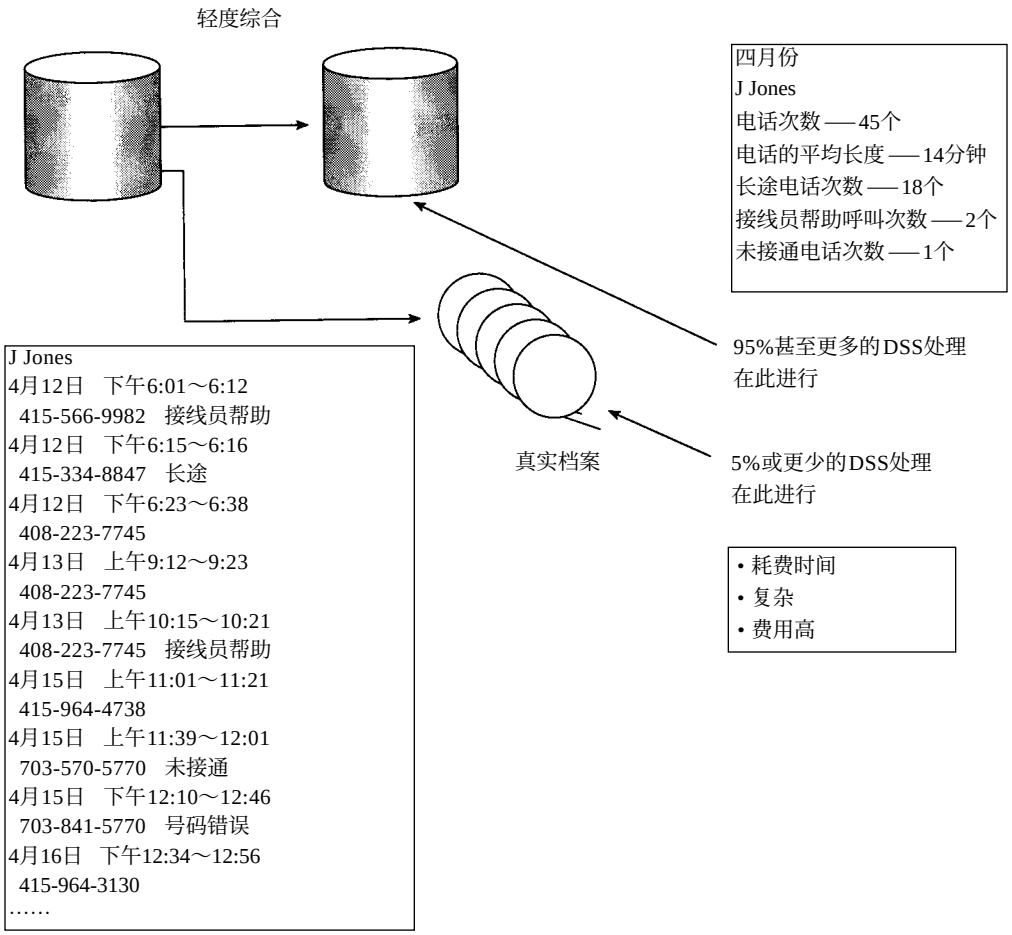


图2-17 双重粒度级可有效地处理绝大多数的请求，并回答任何能够回答的问题。这是最好的并且应作为默认的设计选择

在数据的真实档案层上，存储的所有的细节来自于操作型环境。在这一层上确实有大量的数据。由于数据量太大，因此有必要将数据存放在如磁带这样的介质上。

通过在数据仓库的细节级上创建两种粒度级， DSS设计者可一举两得。大部分 DSS处理是针对被压缩的、存取效率高的轻度综合级数据进行的。如果什么时候需要分析更低的细节级(5%时间或更少的可能)，可以到数据的真实档案层。在粒度真实档案层上，访问数据将是昂贵的、麻烦的和复杂的事情，但如果必须进入这一细节级也只得如此。

随着时间的迁移，如果需要开发某种搜索数据的真实档案级的模式，设计者可能要在轻度综合级上创建某些新数据域。

鉴于费用、效率、访问便利和能够回答任何可以回答的查询的能力，数据双重粒度级是大多数机构建造数据仓库细节级的最好选择。只有当一个机构的数据仓库环境中只有相对较

少的数据时，才应尝试采用数据粒度的单一级别。

2.5 分割问题

分割是数据仓库中数据的第二个主要的设计问题（在粒度问题之后），如图2-11b所示。数据分割是指把数据分散到各自的物理单元中去，它们能独立地处理。在数据仓库中，围绕分割问题的焦点不是该不该分割而是如何去分割的问题。

人们常说，如果粒度和分割都做得很好的话，则数据仓库设计和实现的几乎所有其他问题都容易解决。但是，假如粒度处理不当并且分割也没有认真地设计与实现，这将使其他方面的设计难以真正实现。

当然，数据仓库还有其他设计问题，将在后面章节进行讨论。

2.6 样本数据库

样本数据库是数据仓库的一种有趣的、混杂的形式，它只是真实档案数据或轻度综合数据的子集。术语“样本”源于它是更大数据库的子集(即样本)这一事实，并需要进行定期刷新。图2-18显示了一个样本数据库。

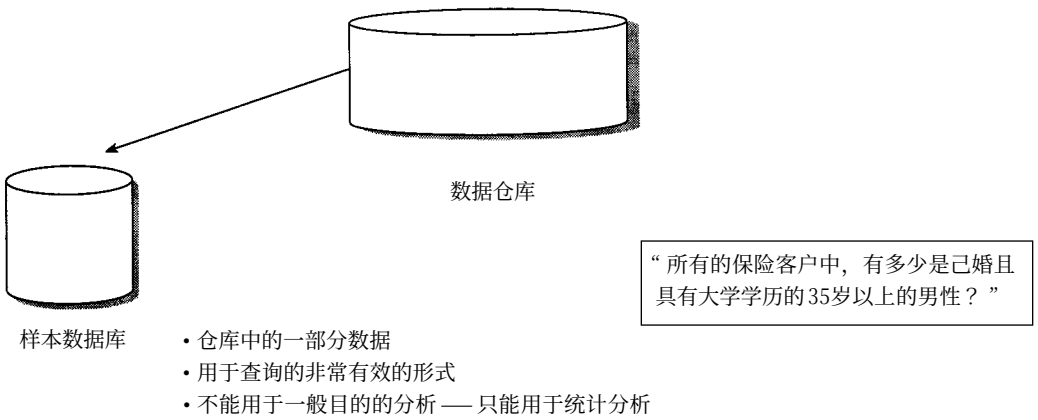


图2-18 样本数据库 —— 另一种改变数据粒度的方法

在某些情况下(如人口统计分析)，样本数据库是非常有用的。但是对使用样本数据库有一些苛刻的限制。除非设计者知道这些限制，否则就不应该创建这种数据库以作为数据仓库的一部分。

样本数据库不是通用的数据库。假如你想知道“ J.Jones是不是顾客？”你就不要在样本数据库中找这条信息。完全可能， J.Jones是一个顾客，但她不在样本数据库的记录中。样本数据库适用于作统计分析和观察发展趋势。当数据必须以整体观察时，样本数据库能提供非常理想的结果，但决不适用于处理单个的数据记录。

现在看一下样本数据库的数据通常是如何载入的？它是用一个程序搜寻一个大规模的数据库，选取1/100或1/1000的记录，然后将这些记录送到样本数据库。于是最终样本数据库的大小将是原先数据库的1/100或1/1000。

样本记录的选取一般是随机的，必要时可采用一个“判断样本”(即记录必须达到一定标

准才能被选中)。判断样本所带来的问题是会使样本数据具有某种偏差，随机抽取数据带来的问题是可能无法进行统计。无论如何，数据是选择作为样本的，所以在样本数据库中找不到任何给定的记录这一事实是说明不了任何问题的。

样本数据库的最大好处是存取效率非常高。即使只是从一个大数据库中抽取了很小的一部分，对它进行访问和分析也相对地有效得多。

换句话说，一个分析员可能花 24小时来浏阅与分析一个大数据库，而浏览与分析一个样本数据库则可能只需 10分钟。在进行启发式分析中，周转时间对可以进行的分析而言是至关重要的。在启发式分析中，分析员运行程序、分析结果、修改程序、再运行程序。如果执行程序就花去24小时，分析和修改程序的过程就会大大地削弱了 (更不用说修改所需的资源)。

但如使用一个10分钟内就足以浏览完的样本数据库，分析员就能很快地完成这种重复过程。总之，DSS分析员的生产效率是由进行整个分析过程的速度来决定的。

有一种说法认为进行统计分析会导致错误的结论。现在来看一种情况：当分析员遇到一个大文件，用325 000 000条记录确定56.7%上路的汽车司机是男人，而使用样本数据库，分析员用了25 000个记录确定55.9%上路的汽车司机是男人。前一种分析比后一种分析需要的资源要大得多，而它们计算出的结论却差别非常非常小。毫无疑问，用大规模数据库进行分析会比较精确，但这种精确的代价实在太高了，尤其在进行启发式处理时，通常要进行循环往复的处理。

如果需要非常高的精确度，行之有效的方法是将要求形式化，并在样本数据库上进行反复处理。这样做，DSS分析员就可较快地将要求形式化。当要求被理解后，再在大规模数据库上仅运行最后一次。

采用样本数据是另一种在数据仓库中改变粒度级以便于进行 DSS处理的方法。

2.7 数据分割

在数据仓库环境中，问题不是要不要对当前细节数据进行分割，而是怎样对当前细节数据进行分割。图2-19说明了数据分割。

对当前细节数据进行分割的总体目的是把数据划分成小的物理单元。数据分割为什么如此重要呢？因为小的物理单元能为操作者和设计者在管理数据时提供比对大的物理单元更大的灵活性。

当数据存放在大的物理单元中时，尤其不能达到：

- 容易重构。
- 自由索引。
- 顺序扫描(若需要)。
- 容易重组。
- 容易恢复。
- 容易监控。

简单地说，数据仓库的本质之一就是灵活地访问数据。如果是大块的数据，就达不到这一要求。因而，对所有当前细节的数据仓库数据都要进行分割。

分割数据的准确含义是什么呢？当结构相同的数据被分成多个数据物理单元时，数据便被分割了。此外，任何给定的数据单元属于且仅属于一个分割。

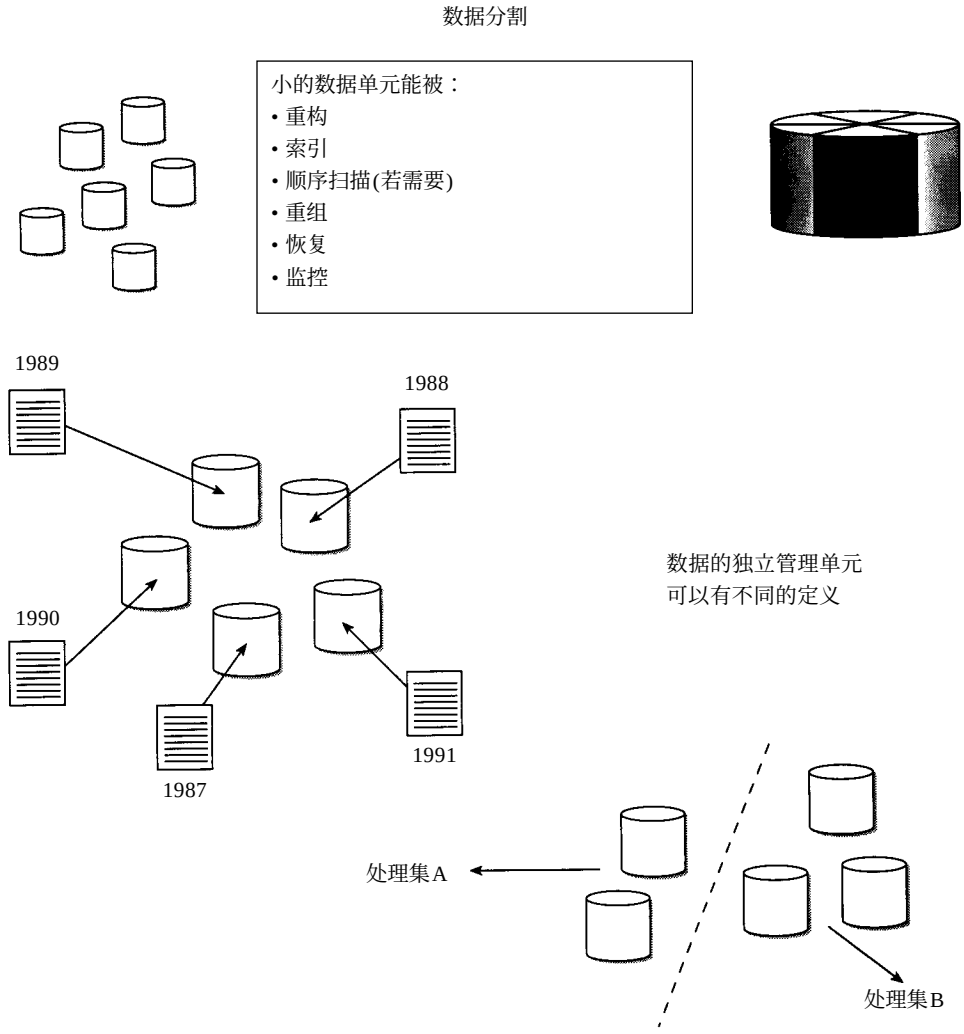


图2-19 独立管理的数据分割可送到不同的处理集，而无须顾及其他的系统考虑

有多种数据分割的标准。例如，按：

- 时间。
- 商业线。
- 地理位置。
- 组织单位。
- 所有上述标准。

数据分割的标准是严格地由开发人员来选择的。然而，在数据仓库环境中，按日期几乎总是分割标准中的一个必然组成部分。

将人寿保险公司如何选择数据分割标准作为一个例子，来看看下列的数据物理单元：

- 1988年健康索赔。
- 1989年健康索赔。

- 1990年健康索赔。
- 1987年人寿保险索赔。
- 1988年人寿保险索赔。
- 1989年人寿保险索赔。
- 1990年人寿保险索赔。
- 1988年意外伤亡索赔。
- 1989年意外伤亡索赔。
- 1990年意外伤亡索赔。

这个保险公司使用了日期即年和索赔类型作为标准来分割数据。

数据仓库开发人员面临的主要问题之一是在系统层上还是在应用层上对数据进行分割。在系统层上进行分割在一定程度上是某些 DBMS和操作系统的一种功能。在应用层上进行分割是由设计的应用程序代码完成的，这是只由开发者和程序员严格地控制的。当在应用层上进行数据分割时，DBMS和系统就不知道一种分割与另一种分割之间的关系。

通常，在应用层上分割数据仓库的数据是很有意义的。这是有某些重要原因的，最重要的是在应用层上每年的数据可以有不同的定义。 1988年和1989年的数据定义，可以相同也可以不相同。仓库中数据的性质是长期数据积累的结果。

当数据在系统层上分割时， DBMS不可避免地希望只有一种数据定义。假定数据仓库中保存的数据时间较长(如达到十年)，而且数据定义经常变化，让 DBMS或操作系统去管理一个本该只有一种数据定义的系统将是毫无意义的。

在应用层上管理数据分割的另一重要特点是它能从一个处理集转移到另一个处理集而没有损失。在数据仓库环境中，当工作负载和数据量成为真正的负担时，这种特点就是一种真正的优点。

对数据分割最严峻的检验是提出这样的问题：“能否在分割中加入索引而不会明显地妨碍其他操作？”如果一种索引能随意加上去的话，那么这种分块就足够理想了。如果索引还不能很容易地加入，那么这个分块还得分得更精细一点。

2.8 数据仓库中的数据组织

迄今为止，我们还没有详细研究数据仓库中所建立的数据结构是怎样的。数据仓库中有多种数据组织形式，我们将讨论几类比较常见的结构。

数据仓库中最简单最常用的数据组织形式也许是简单堆积结构，如图 2-20所示。

图2-20表示了从操作型环境中取出每天的事务处理，然后综合成数据仓库记录，这个综合可根据顾客、帐目或者任何组织到数据仓库的主题领域来进行。这里的事务处理是以天来进行综合。换句话说，对一个顾客的一个帐号的每天的所有活动进行合计，并在一天一天的基础上输入数据仓库。

图2-21表示简单逐日堆积数据的一种变种，称之为轮转综合数据存储。

数据用与前面相同的处理方法从操作型环境输入到数据仓库环境中。只是在轮转综合文件中的数据才被输入到不同的结构形式中。第一周的七天中的活动被逐一综合到七个每日相应的位置，到第八天，将七个每日位置的数据加到一起，并放入第一周的数据位置中。然后，第八天的每日总计加到第一个每日数据位置。

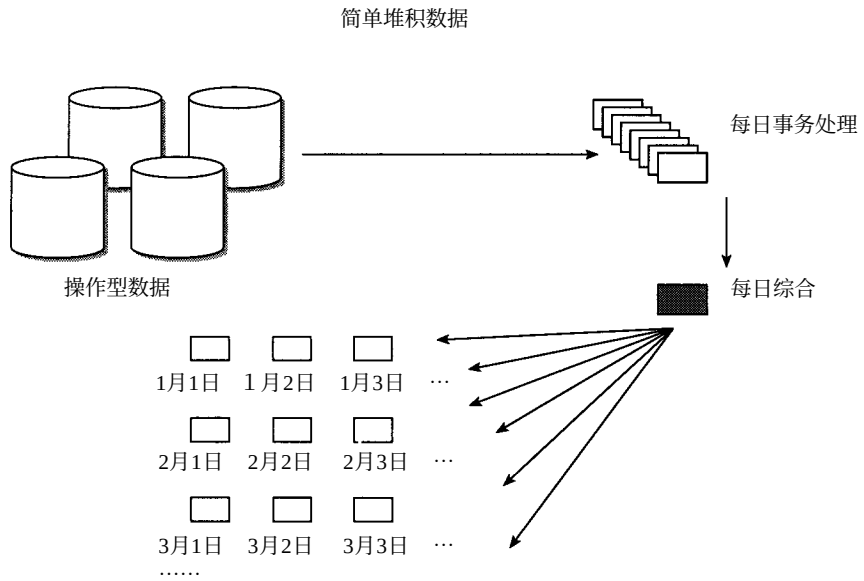


图2-20 数据仓库中最简单的数据组织形式是以逐个记录为基础的数据堆积 —— 简单堆积数据

月底将每周位置的数据加到一起，并放入第一个每月相应的数据位置处，然后每周数据位置清零。到了年底，将每月位置数据加到一起，放入第一个年度相应的数据位置处，然后每月数据位置清零。

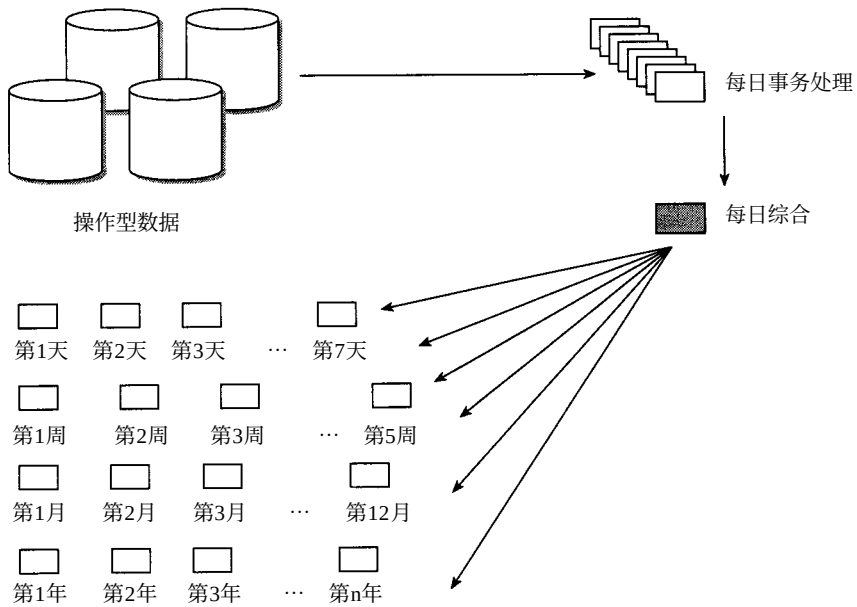


图2-21 轮转综合文件是简单堆积文件的变种

轮转综合数据结构与数据的简单堆积结构相比，仅处理非常少的数据单元。它们之间的优缺点比较如图2-22所示。

数据仓库数据的另外一种组织形式是简单直接文件，如图 2-23所示。

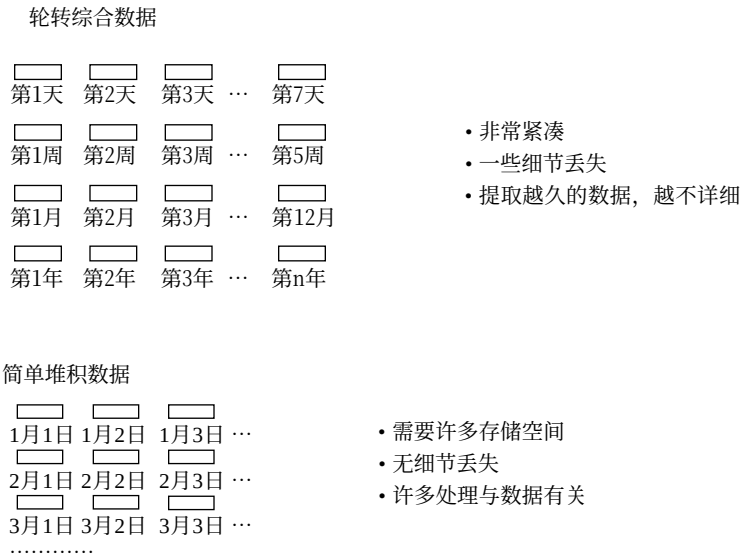


图2-22 轮转综合数据与简单堆积数据的比较

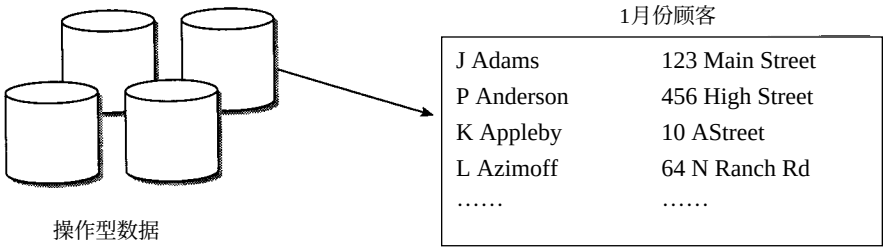


图2-23 简单直接文件——另外一种数据仓库的结构

图2-23表明，数据仅仅是从操作型环境拖入数据仓库环境中，并没有任何累积。另外，简单直接文件不是在每天的基础上组织的，而是以较长时间为单位的，比如一个星期或一个月。因此，简单直接文件是间隔一定时间的操作型数据的一个快照。

依据两个或更多的简单直接文件能生成一种连续文件。图 2-24是把1月份和2月份的两个数据快照合并，创建数据的一个连续文件。连续文件中的数据代表从第一个月到最后一个月的连续数据。

当然，连续文件也可以通过把一个快照追加到一个以前生成的连续文件上来创建，如图2-25所示。

数据仓库中有许多其他的数据组织形式，最常用的是：

- 简单堆积。
- 轮转综合。
- 简单直接。
- 连续。

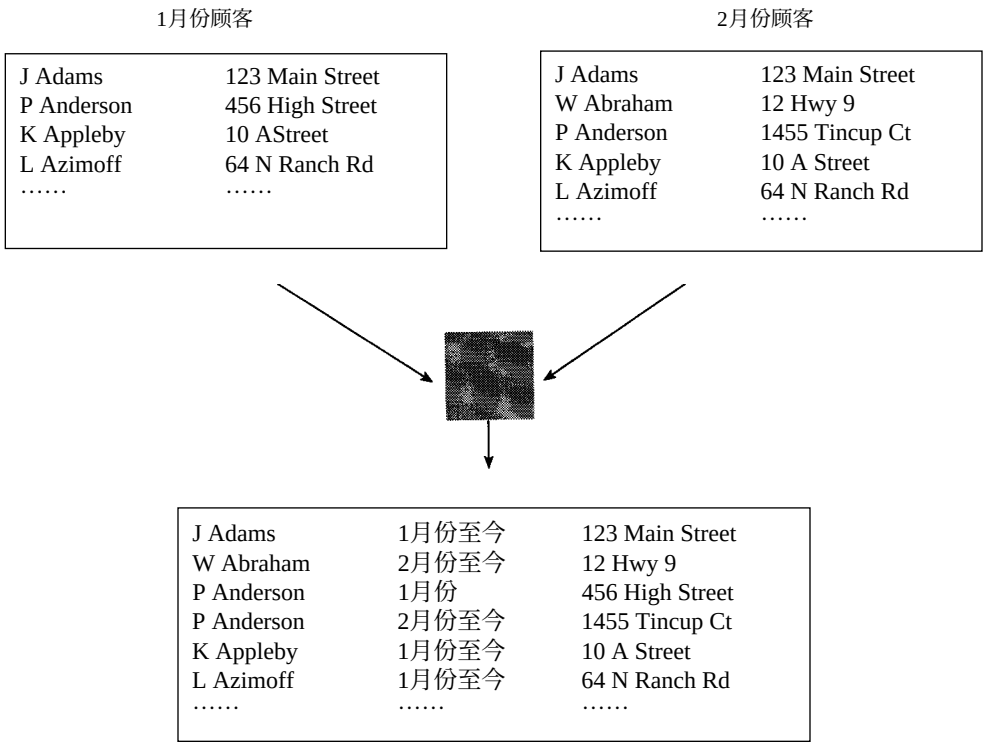


图2-24 从直接文件创建一个连续文件

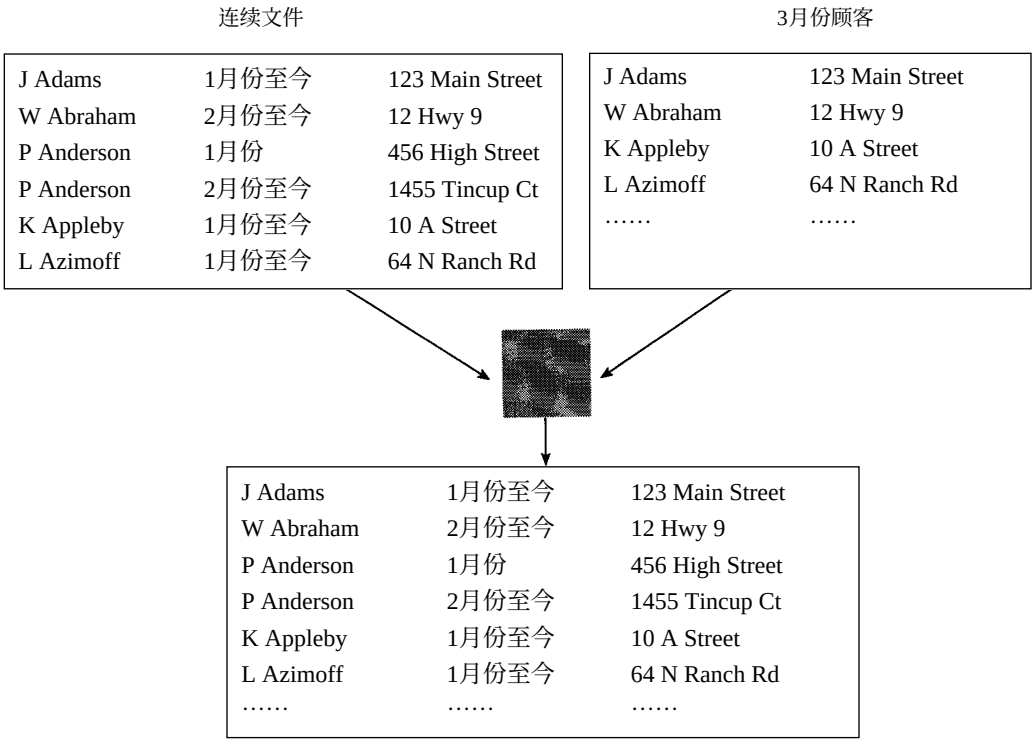


图2-25 由简单直接文件创建连续文件，或把简单直接文件追加到连续文件

在键码层，数据仓库的键码不可避免地是复合键码，这有两种强制性的理由：

- 日期——年、年/月、年/月/日，等等，几乎总是键码的一部分。
- 因为数据仓库中的数据是分割的，分割的不同分量表现为键码的一部分。

2.9 数据仓库——标准手册

数据仓库与许多人(管理人员、DSS分析员、开发人员、计划人员等)相关。而在大多数机构中，数据仓库还是个新事物。于是就需要有一个官方组织对数据仓库进行解释和说明。

称数据仓库说明为“标准手册”可能有点死气沉沉。标准手册里是一些枯燥的含义，并以无人问津和布满灰尘而闻名。但是以某种内部形式发行还是很有价值的尝试。

这种出版物应该包括下列内容：

- 描述什么是数据仓库。
- 描述对数据仓库输送信息的源系统。
- 如何使用数据仓库。
- 有了问题如何获得帮助。
- 谁负责什么。
- 仓库的迁入计划。
- 仓库数据如何与操作型数据相关联。
- 如何为DSS使用仓库数据。
- 什么时候不往仓库中加数据。
- 仓库里没有什么类型的数据。
- 可利用的元数据的指示。
- 记录系统是什么。

2.10 审计和数据仓库

伴随数据仓库出现的一个有趣的问题是：是否能够或是否应该对数据仓库进行审计？答案是能对数据仓库进行审计；已有进行详细审计的几个例子。然而有更多的理由表明，即使能对数据仓库进行审计，也不应该从中进行审计。不进行这种审计的主要原因如下：

- 原先在数据仓库中没有的数据会突然出现。
- 当需要审计能力时，数据进入数据仓库的定时会发生急剧变化。
- 当需要审计能力时，数据仓库的备份和恢复限制会发生急剧变化。
- 在仓库中审计数据，会使仓库中数据的粒度处在一个非常的级别上。

总之，在数据仓库环境中进行审计是可能的。然而，审计带来的复杂性使得审计在其他地方进行更有意义得多。

2.11 成本合理性

数据仓库的一个有趣的问题是：数据仓库的成本合理性通常不是在先验的投资回报率(ROI)基础上来进行判断的。要做这样的分析，必须在创建数据仓库之前就知道数据仓库带来的收益。

通常情况下，在创建数据仓库时，实际收益是不知道的甚至是无法预测的，因为数据仓

库的使用方式完全不同于其他信息系统所构造的数据和系统。数据仓库的使用是用一种与其他信息处理不同的模式进行的，有点像“告诉我，你想要的是什么，然后我才能告诉你你真正想要的是什么”这种模式。直到数据仓库的首次循环设计过程完成并可以利用以前， DSS分析员不能真正说出数据仓库的可能性和潜力有多大。一旦 DSS分析员插手仓库，他或她就能开始揭示DSS处理的潜力。

因此，经典的ROI技术不能应用到数据仓库环境中。幸运的是，数据仓库是渐进式地建立的。第一次循环设计过程能很快完成，并且只需相对较少的费用。一旦数据仓库的第一部分已经建立并载入数据，分析员就能开始研究可能性。只有在这个时候，分析员才能着手证明仓库开发费用的合理性。

根据经验，数据仓库第一次循环的设计规模要适中，小到足以建成仓库，大到仓库是有意义的。所以，数据仓库最好一次构造一个小规模的循环设计，并且在仓库开发人员与 DSS分析员之间适当而且快速地建立直接反馈回路。

此外，这些理由也说明，为什么说若最初设计的数据仓库 50%是精确的，那么设计就是成功的。根据仓库开发人员与 DSS分析员之间的反馈回路将不断地修改现有的仓库数据，并向仓库中追加其他新数据。

数据仓库典型的首选方向集中于以下这些主题领域之一：

- 金融。
- 市场。
- 销售。

有时，数据仓库的首选主题领域会专注于：

- 工程/制造。
- 保险行业。

2.12 清理仓库数据

数据并非只是注入数据仓库，它在数据仓库中也有自己的生命周期。到了一定时候，数据将从仓库中清除。数据清理问题是数据仓库设计人员无法回避的基本设计问题之一。

从某种意义上讲，数据根本不是从数据仓库中清除，而仅仅是上升到更高的综合级。数据清理或数据细节转化主要有以下几种方式：

- 数据加入到失去原有细节的一个轮转综合文件中。
- 数据从高性能的介质(如DASD)转移到大容量介质上。
- 数据从系统中实际清除。
- 数据从体系结构的一个层次转到另一个层次，比如从操作型层次转到数据仓库层次。

因而，在数据仓库环境之中有种种数据清理或者转化的方式。数据的生命周期 (包括清除或最终档案转移)应该是数据仓库设计过程中活跃的部分。

2.13 报表和体系结构设计环境

如果说一旦数据仓库建立起来，所有的报表和信息处理都将在此实现，这只是一种诱惑。情况确实不是这样。恰如其分地应归属于操作型系统领域的，是一个正统的处理类别。图 2-26表明不同的处理风格应位于什么位置。

图2-26显示，操作型报表是为办事层用的，并且焦点基本在行式项目上。数据仓库或信息型处理的焦点在管理，包含综合或其他情况下的计算信息。在数据仓库中一旦基本数据计算已经完成，报表格式很少使用行式项目、细节信息。

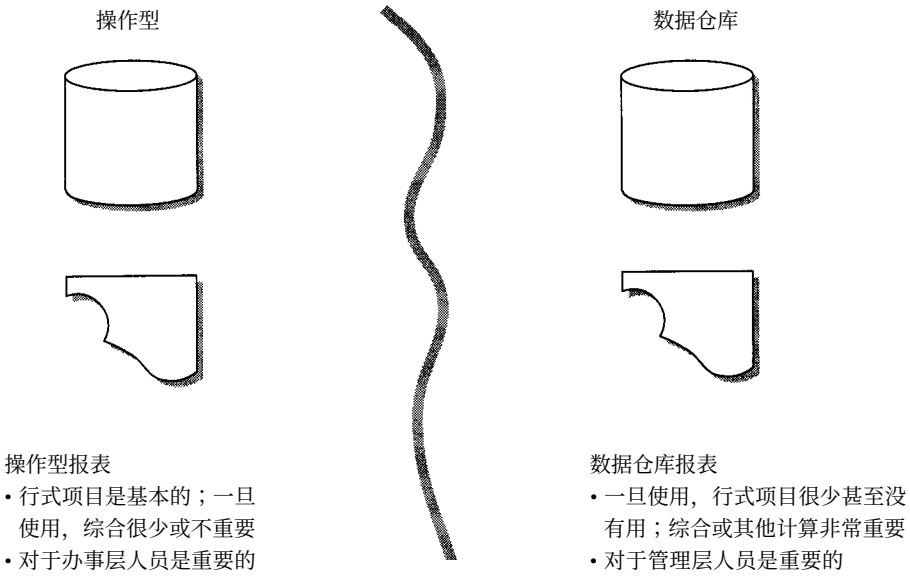


图2-26 两种报表的区别

2.14 机遇性的操作型窗口

数据仓库是DSS处理的基础，包含的都是档案信息，而且大部分是至少 24小时以前的。但是档案数据在体系结构设计环境中别的地方能找到，特别是在操作型环境中也能找到。

在数据仓库中，发现5～10年这样漫长的档案数据是很常见的。由于时间范围的缘故，数据仓库中存在有大量的数据。

在操作型环境中发现的档案数据的时间范围称之为数据的“操作型窗口”，它一般不很长，只能从一个星期到两年。但是操作型环境中档案数据的时间范围，并不是档案数据在操作型环境和在数据仓库环境的唯一区别。不同于数据仓库，操作型环境的档案数据不是大宗的，并具有高访问率。

为了理解在操作型环境中刷新的、非大宗的、高访问率的档案数据的作用，考虑一家银行的工作方式。在一个银行环境中，顾客有理由想要找到有关这个月事务处理的信息，如这个月的租金支票清了没有？什么时侯进行的支票支付存款？这个月的结余是多少？银行是否取钱交了上星期电费？

可以要求把非常细节的和新近的事务处理 (仍然是档案的)包含在银行的操作型环境中。但是指望银行告诉顾客“ 5年前是否给杂货店开过一张支票？”或“ 10年前一张竞选捐款支票是否兑换成现金了？”这是不是合理的呢？这些处理很难在银行的操作型系统领域进行，不仅因为这些处理已时间很久了，而且它们包含的数据具有非常低的访问率。

时间操作型窗口从一种行业到另一种行业是变化的，甚至一个行业内部的数据和活动类型也是变化的。例如，一个保险公司可能有一个 2到3年这样漫长的操作型窗口。至少与其他类型

行业相比，保险公司内部的事务处理率是很低的，在顾客和保险公司之间直接交互相对就更少。相反，银行活动的操作型窗口非常短，一般从0到60天，银行有许多与顾客之间的直接交互。

一家公司的操作型窗口由该公司属于什么类型的行业来决定。如果是个大公司，它可能拥有不止一个操作型窗口，这是由所处理业务的细目决定的。例如，在一家电话公司里，顾客使用数据可能拥有30到60天的操作型窗口，而卖主/供应商活动可能拥有2到3年的窗口。

下面是不同类型行业档案数据操作型窗口的一些建议：

- ■ 保险公司： 2～3年
- ■ 银行信托处理： 2～5年
- ■ 电话公司
  - • 顾客使用： 30～60天
  - • 供应商/卖主活动： 2～3年
- ■ 普通银行
  - • 顾客帐目活动： 30天
  - • 卖主活动： 1年
  - • 贷款： 2～5年
- ■ 零售业
  - • SKU活动： 1～14天
  - • 卖主活动 1周～1个月
- ■ 航空公司
  - • 航班座位活动 30～90天
  - • 卖主/供应商活动 1～2年
- ■ 公用事业
  - • 顾客使用 60～90天
  - • 供应商活动 1～5年

操作型窗口的长度对DSS分析员而言是很重要的，因为它决定了分析员在哪里进行不同的分析和能做什么类型的分析。例如， DSS分析员能对在操作型窗口里找到的数据进行单项分析，而不能做大的趋势分析。操作型窗口的数据适于高效单个访问，只有当数据传出操作型窗口时，才适于进行大量数据的存储和访问。

另一方面， DSS分析员能对在操作型窗口外找到的数据进行全盘趋势分析。在操作型窗口之外的数据能被整体地访问和处理，而访问任何一个单个数据单位都是不理想的。

2.15 小结

本章叙述了数据仓库的面向主题，每个数据记录的随时间变化，以及通过公共键码的连接。

能做出的两个最重要的设计决策是数据粒度和数据分割。对大多数机构而言，双重粒度是很有意义的。数据分割通常是在应用层上而不是在系统层上将数据分成小的物理单元。