第4章　数据仓库中的粒度

数据仓库开发者需要解决的最重要的单一设计问题是数据仓库中的粒度确定。当数据仓库的粒度合理确定后，设计和实现的其他问题就会非常容易地解决，相反如果没有合理地确定粒度的话，就会影响其他每个方面。

对选择合适级别的粒度的权衡——像我们在以前已讨论过的——将集中讨论在高粒度级别上管理大量数据和存储数据，数据的细节的用法将不被考虑。

4.1　粗略估算

确定合适的粒度级的起点，是粗略估算数据仓库中将来的数据行数和所需 DASD(直接存取存储设备)数。毫无疑问，即使在最好的情况下我们也仅能做一下估计。但在建立数据仓库之初，所需的只是一个数量级上的估计。

有一个计算数据仓库所占的空间的算法，如图4-1所示。第一步是确定数据仓库中将要创建的所有表。然后，估计每张表中的行的大小。确切大小可能难以知道，估计一个下界和一个上界就可以了。

接下来，估计一年内表中的最少行数和最多行数。这是设计者所要解决的最大问题。比方说一个顾客表，就应该估计在一定的商业环境和该公司的商业计划影响下的当前的顾客数；如果当前没有业务，就估计为总的市场业务量乘以市场份额；如果市场份额不可知的话，就用竞争对手的业务量来估计。总之，要从一方或多方收集顾客的合理估算信息开始。

如果数据仓库是用来存放业务活动的话，就要估计顾客数量，以及估计每个时间单位内业务活动量。同样，可用相同的方法分析当前的业务量、竞争对手的业务量、经济学家的预测报告，等等。

估计数据仓库环境中的行数/空间大小
1. 对每一个已知的表：
　　计算一行所占字节数的
　　－最大估计值
　　－最小估计值
对一年内：
　最大行数可能是多少？
　最小行数可能是多少？
对五年内：
　最大行数可能是多少？
　最小行数可能是多少？
对表的每个键码：
　该键码的大小(按字节)是多少？
一年总的最大空间=最大行大小×一年内最大行数
一年总的最小空间 =最小行大小×一年内最小行数
　累加索引空间
2. 对所有已知的表重复第1步。

图4-1　空间/行数计算

一旦估计完一年内数据仓库中数据单位的数量(用上下限推测的方法)，就用同样的方法对五年内的数据进行估计。

粗略数据估计完后，就要计算一下索引数据所占的空间。对每张表(对表中的每个键码)确定键码的长度和原始表中每条数据是否存在键码。

现在将各表中行数可能的最大值和最小值分别乘以数据的最大长度和最小长度。另外，还要将索引项的数目与键码的长度的乘积累加到总的数据量中去。

4.2 粒度划分过程的输入

可以将估计的行数和DASD数作为粒度划分过程的输入，如图 4-2所示。此时，数量级的精确性才是最重要的。

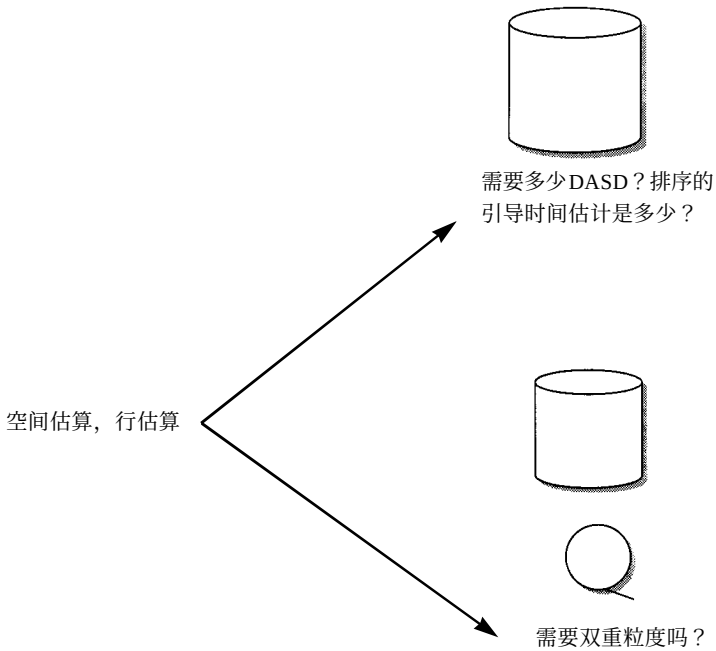


图4-2 使用空间估计的结果

4.3 双重或单一的粒度？

一旦上面的估计完成后，下一步就要将数据仓库环境中总的行数和图 4-3中所示的表格进行比较。根据数据仓库环境中将具有的总的行数的大小，设计和开发必须采取不同的方法。以一年期为例，如果总的行数小于 10 000行，那么任何的设计和实现实际上都是可以的。如果一年期总行数是100 000行或更少，那么设计时就需小心谨慎。如果在头一年内总行数超过1 000 000行，那么就要请求采取双重粒度级。如果在数据仓库环境中总行数超过 10 000 000行的话，必须强制采取双重粒度级，并且在设计和实现中应该小心谨慎。

对于五年期，行的总数大致依据数量级改变。对五年以后的推测是：

- 在管理数据仓库中的大量数据时，将有更多的专门技术可用。
- 硬件费用有所下降。
- 可以使用功能更强大的软件工具。
- 最终用户更加专业化。

所有的这些因素表明在一段长的时间内可以管理更大的数据量。

有趣的一点是，数据仓库存储时总的字节数与数据仓库的设计和粒度是几乎没有关系的。换句话说，记录是 25个字节长或250个字节长是没有关系的。图 4-3所示的表仍旧可用。原因与数据的索引有关，不论被索引的记录的大小，都需要同样数量的索引项。只有在一

些特殊情况下，被索引的记录的实际大小才影响决定数据仓库是否采用双重级粒度的策略。

一年期	
10 000 000	双重粒度级且认真设计
1 000 000	双重粒度级
100 000	认真设计
10 000	实际上任何设计都行

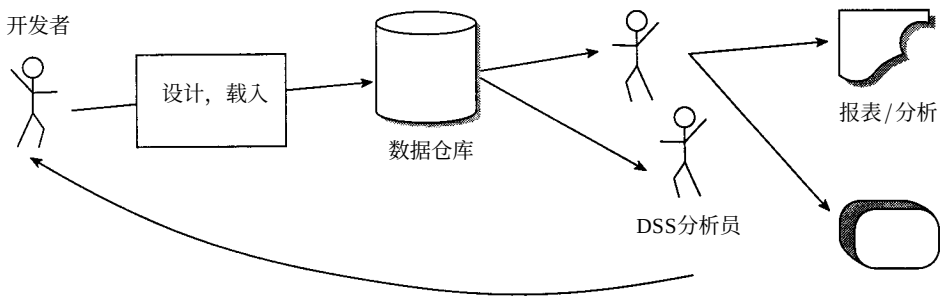
五年期	
20 000 000	双重粒度级且认真设计
10 000 000	双重粒度级
1 000 000	认真设计
100 000	实际上任何设计都行

图4-3 粒度的阈值

4.4 确定粒度的级别

完成了一些简单分析之后(事实上，这时大多数公司发现他们需要双重粒度)，下一步就要精确地确定粒度的级别。开始时是需要一些常识和直觉的。在很低的细节级上建立轻度汇总的数据级是没有意义的，因为需要太多的资源来处理数据。而在太高的细节级上建立轻度汇总的数据级，则意味着许多分析必须在真实档案级上进行。因此确定轻度汇总的粒度级的第一件事是进行有根据的猜测。

但进行有根据的猜测也只是一个开端。还需要一定数量的反复分析来改进这个猜测，如图4-4所示。对于轻度汇总的数据为了确定合适的粒度级别，唯一可行的方法是将数据拿到最终用户的面前。只有当最终用户实际看到了数据之后，我们才能作出确定的回答。图 4-4说明了我们所需做的反复的循环。



经验规则：在第一次的设计周期中，如果50%的工作是正确的，那么整个设计就是成功的

- 快速 建立数据仓库的很小的子集并认真听取用户的反馈意见。
- 用原型法。
- 看看别人做了些什么。
- 找一个有经验的用户协同你工作。
- 看看机构现在已经有了些什么。
- 用模拟的输出进行JAD(联合应用程序设计)会议。

图4-4 最终用户的态度：“既然我看到了我能够做些什么,我就能告诉你什么是真正有用的。”

4.5 一些反馈循环技巧

我们可以用以下的一些技巧来使反馈循环成为一个和谐的循环：

- 用很小而很快的步伐建立数据仓库最初的几个部分，仔细聆听最终用户的意见。随时准备做快速的调整。
- 如果可以使用原型工具的话应用原型法，并使用从原型中收集的观察结果而使反馈循环起作用。
- 看看别人是怎样确定他们的粒度级别，学习一下他们的经验。
- 与一个对整个过程了解的有经验的用户一起进行反馈的处理。不论什么时候都让你的用户在暗中作为反馈循环的动力。
- 看看本机构现在有什么系统正在运转。
- 进行JAD会议并模拟其输出以得到想要的反馈。

有好多方法用来提高数据的粒度，如以下所列：

- 当源数据置入数据仓库时，对它进行汇总。
- 当源数据置入数据仓库时，对它求平均或进行计算。
- 把最大/最小的设定值置入数据仓库。
- 只把显然需要的数据置入数据仓库。
- 用条件逻辑选取记录的一个子集置入数据仓库。

对于数据怎样轻度汇总是没有限制的(限制只存在于设计者的脑海里)。

有一点很重要，在典型的需求系统的开发中，在还不清楚大部分需求之前就忙于进行是不明智的。但在数据仓库的建造中，如果已知了至少一半的需求后，还不开始同样也是不明智的。换句话说，在建造数据仓库中，如果开发者想等着大多数需求明了后才开始工作，那么这个仓库是永远建不起来的。尽快启动与DSS分析员的反馈循环是非常重要的。

4.6 粒度的级别——以银行环境为例

以如图4-5所示的一个银行或金融环境的简单数据结构为例。

在左边(在操作级上)是操作型数据。在这里我们可以看到银行业务的详细数据。六十天的活动数据都存储在操作型的联机环境中。

操作型数据的右边是轻度汇总级的数据，总共是十年的活动记录。对于给定帐户，特定月份的活动记录存储在数据仓库的轻度汇总部分。由于记录很多，所以这部分比原记录更加简洁。在轻度汇总数据级有更少的DASD数和更少的行。

当然，也有真实档案级的数据，其中存储着每一个细节的记录。真实档案级的数据存储在适合于大量数据管理的媒体上。请注意并不是数据的所有字段都传送到真实档案级中去，只有那些由于有合法的理由、信息性理由等所需要的域才被存储。那些即使在档案模式中以后有用的数据在传送到真实档案级时也会从系统中清除出去。

真实档案级数据可以存储在如磁带机这样一种单一的媒体上。磁带机作为存储介质非常便宜，但作为访问介质是非常昂贵的。然而，我们可以将有可能需要访问的一小部分的真实档案级数据联机存储，这是完全可能的。例如，一个银行可以将最近 30天的业务记录联机存储。最近30天的数据是真实档案数据，而它们仍旧是联机的。 30天结束后，这些数据被送到磁带机上，腾出的空间就可以存放下一个30天的真实档案数据。

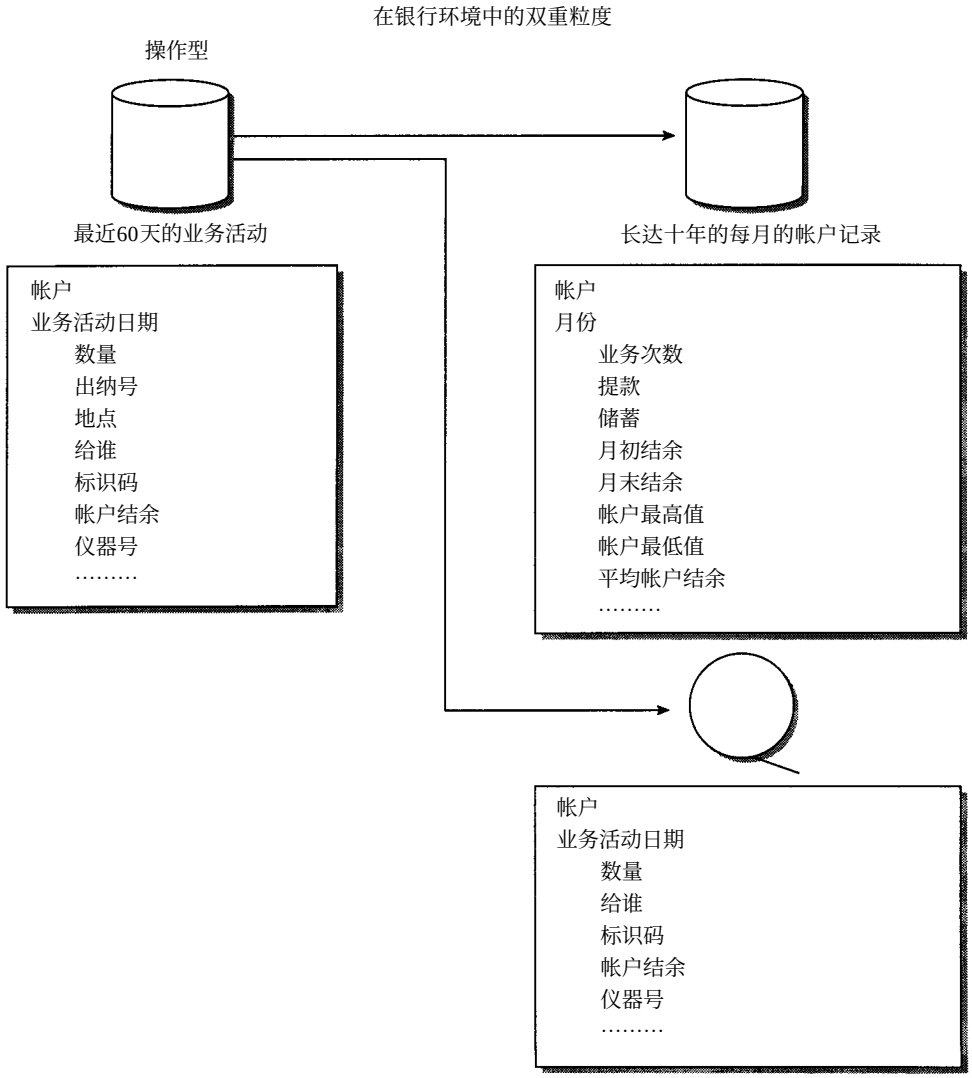


图4-5 在银行环境中一个双重粒度的简单例子

现在考虑银行/金融系统体系结构设计环境中数据的另一个例子，如图 4-6所示。

图4-6显示了整个环境中客户的记录。操作型环境中显示了当前使用的准确的客户数据，在轻度汇总级存放的也是同样的数据(通过数据的定义)，但只是一个月中某个时刻的快照。

还有一个连续的文件存放长时间范围的数据(过去 10年的数据)，它是从每月文件中产生的。用这种方式，一个客户的历史记录能被追溯到很长一段时间。

再来看另一类企业 —— 制造业中体系结构设计环境的一个例子，如图 4-7所示。

在操作级上是完成若干给定部件的组装的生产线的记录。随着生产线的运转，每一天都会积累许多记录。

轻度汇总级包括两个表，一个汇总某一部件一天中的生产情况，另一个汇总生产线上的生产情况。部件生产累计表存放的是90天内的数据，而组装表只存放一天汇总的的生产情况数据。

真实档案级的数据包括每个生产活动的详细记录。而在银行系统中只有以后有用的数据

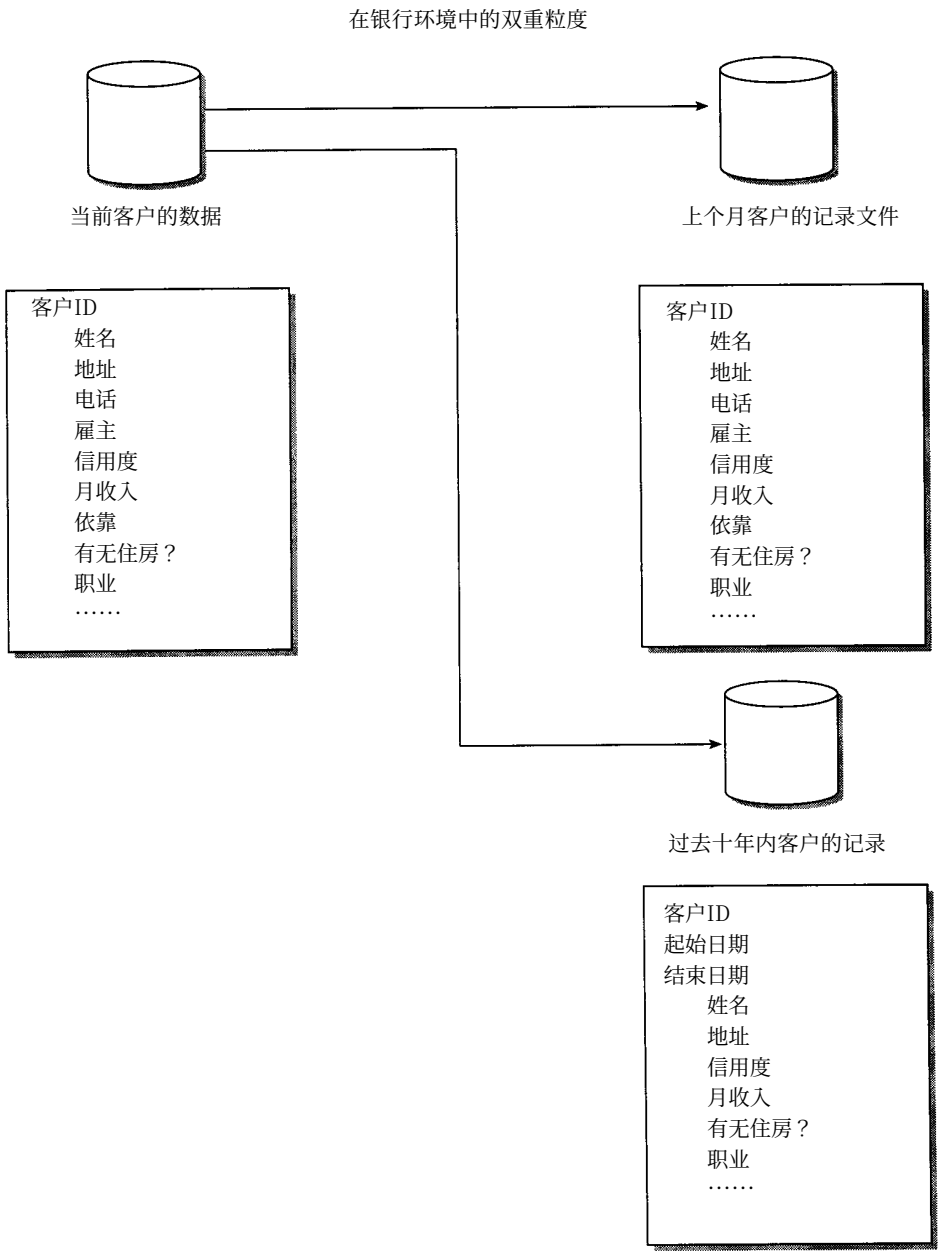


图4-6 银行环境中双重粒度的另一种形式

字段才被存储(事实上是以后有可能有用的字段)。

图4-8所示是制造业环境中另一个关于数据仓库粒度的例子。在操作型环境中有一个活动订单文件，存储所有需要生产活动的订单。数据仓库中存放的是 10年内的订单历史。订单历史表有一个主键码和多个辅键码。只有以后用于分析的数据才存储在数据仓库中。但订单数量太少，没有必要置入真实档案级。当然，如果开始出现过多的订单，那么置入一个较低的粒度级也可能是必要的。

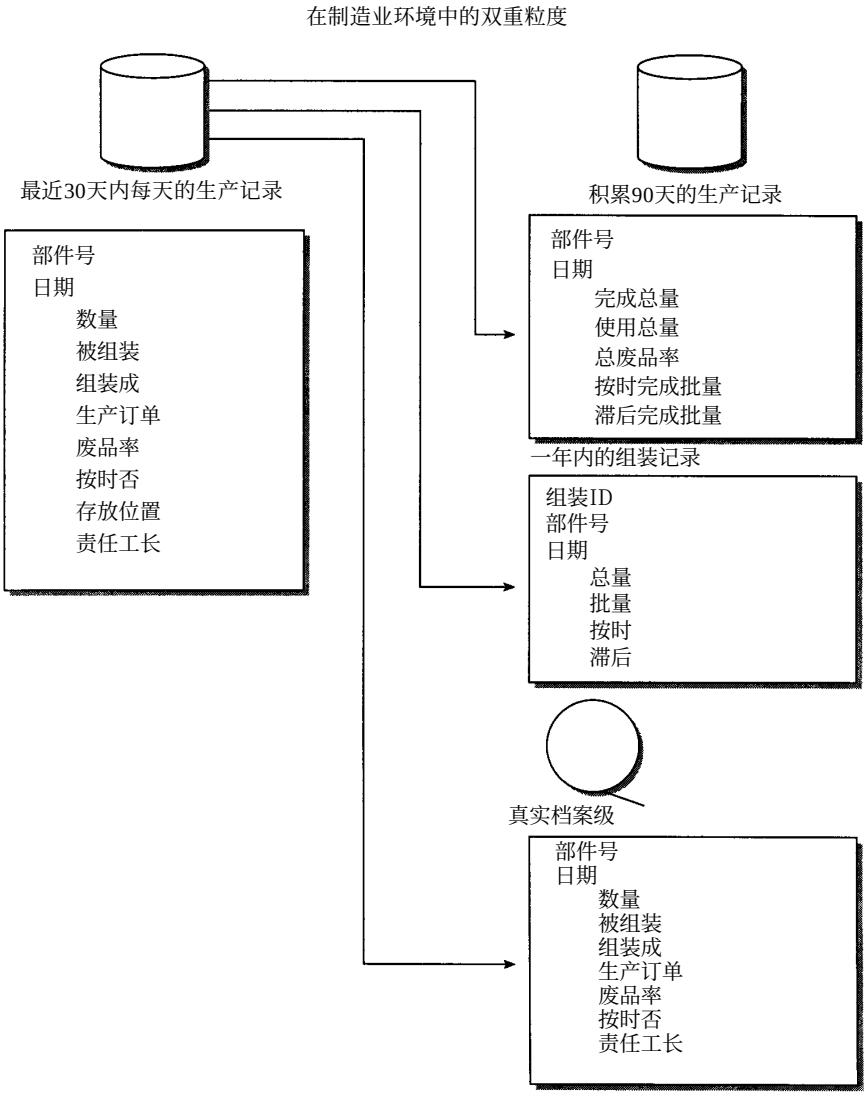


图4-7 制造业环境中的某些不同粒度级别

请看另外的一个例子，如图 4-9所示，这是一个适合保险公司体系结构设计环境中数据的粒度转换。保险费支付信息收集在一个活动文件中。然后过一段时间，这些信息传送到数据仓库中。因为这里的数据相对较少，所以用双重粒度是没有必要的。然而，由于保险费交付的定期性特点，交付数据是作为一个数组的一部分存放在数据仓库中的。

看一下如图4-10所示的保险业环境中体系结构的另一个例子，注意它的保险索赔信息。

在当前的索赔系统中(环境的操作型部分)，存储了索赔的大量详细数据。当一个索赔已解决(或已确定不予解决)，或者索赔隔了好长时间还未办理，这个索赔的信息就被传送到数据仓库中去。在传送时，索赔信息用多种方式汇总 —— 按每个代理在每个月，按每个月索赔的类型，等等。在一个较低的详细级别上，索赔信息在真实档案级上无限期地保存着。在数据传送到真实档案级的其他情况下，只有那些以后有可能用到的数据才保留下来(它们是在操作型

环境中找到的大部分信息)。

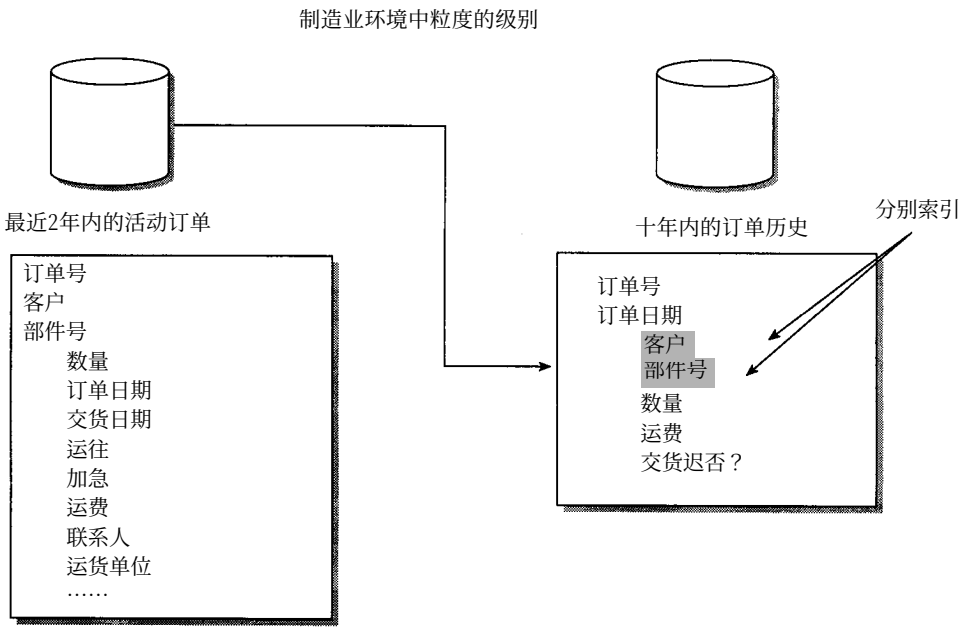


图4-8 订单太少，不需要双重粒度

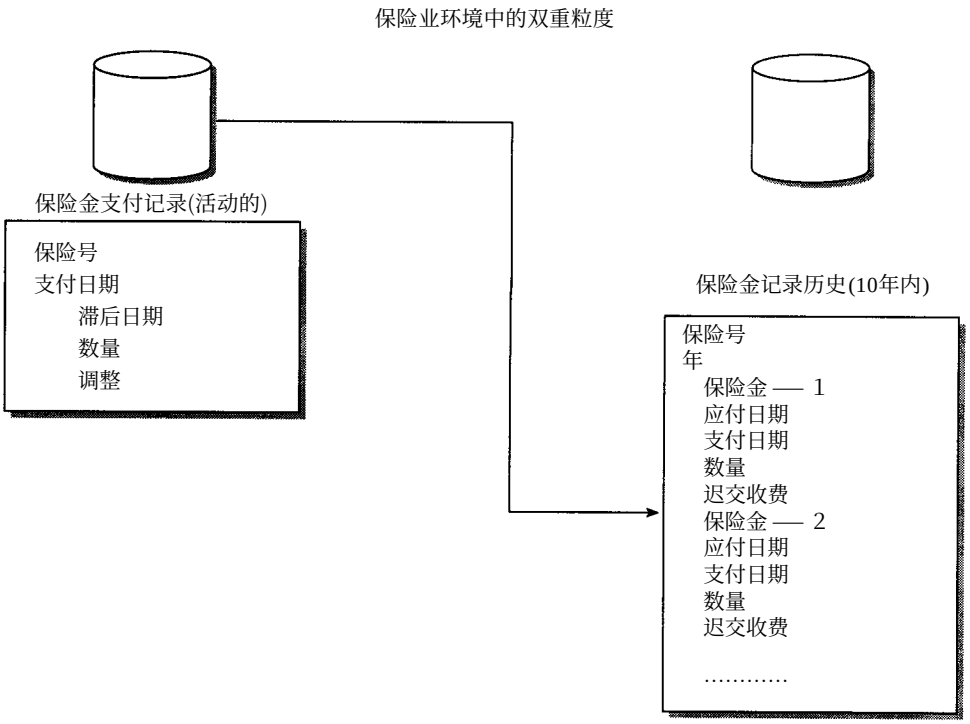


图4-9 保险金支付记录的数量很少，故没有必要用双重粒度，并且由于允许保险金记帐，就有机会来创建一个数据组

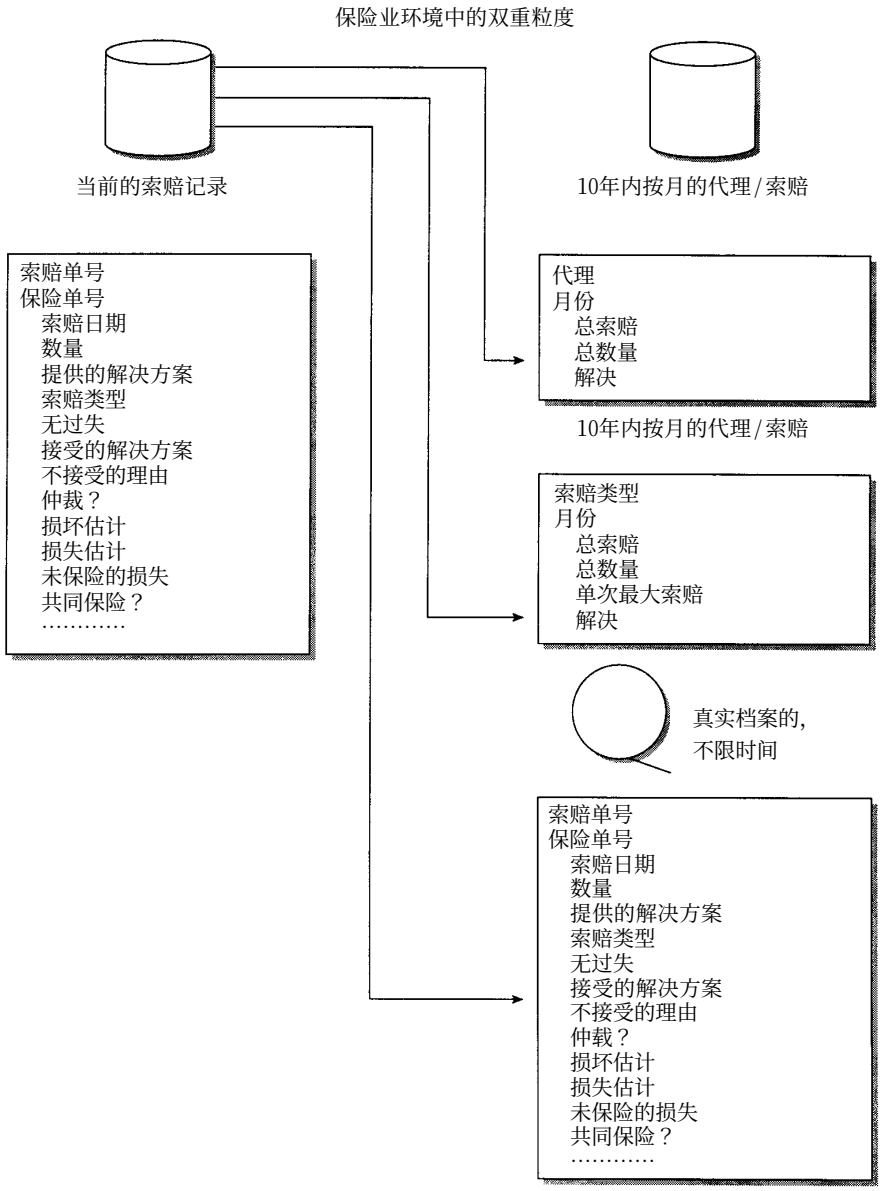


图4-10 在数据仓库的轻度汇总部分存放了按非主键码汇总的索赔信息。索赔信息必须作为数据仓库体系结构中的真实档案部分无限期地存放

以上就是在不同行业中粒度和体系结构设计环境的一些例子。

4.7 小结

选择合适的粒度级别是体系结构设计环境成功的关键。选择粒度级别的一般方法，是利用常识，建立数据仓库的一小部分，并让用户去访问这些数据。然后仔细聆听用户的意见，根据他们的反馈意见适当调整粒度的级别。

最坏的想法是想要事先设计好所有的粒度级别，再进行数据仓库的建造。即使在最好的情况下，能使设计的50%是正确的就已经很不错的了。数据仓库环境的特点就是只有当决策支持系统分析员实际看到了报告之后，才能想像哪些是真正需要的。