

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利说明书

专利号 ZL 98809341.3

[45] 授权公告日 2008年6月4日

[11] 授权公告号 CN 100392643C

[22] 申请日 1998.7.7 [21] 申请号 98809341.3

[30] 优先权

[32] 1997.7.21 [33] SE [31] 9702763-5

[86] 国际申请 PCT/SE1998/001335 1998.7.7

[87] 国际公布 WO1998/004351 英 1999.1.28

[85] 进入国家阶段日期 2000.3.21

[73] 专利权人 MYSQL 股份公司

地址 瑞典乌普萨拉

[72] 发明人 U·M·隆斯特伦

[56] 参考文献

US 5410703A 1995.4.25

JP 8202597A 1996.8.9

EP 0449494A2 1991.10.2

审查员 洪岩

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 王勇 王忠忠

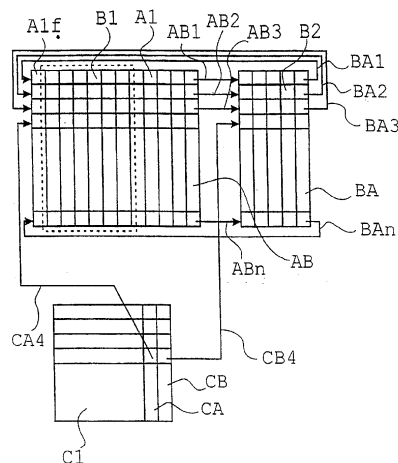
权利要求书 3 页 说明书 16 页 附图 4 页

[54] 发明名称

数据库的模式改变方法

[57] 摘要

本发明涉及执行数据库中模式改变的方法，其中模式改变是复杂模式改变，诸如复杂软模式改变，它导致在该模式改变前的第一模式版本与在该模式改变后的第二模式版本兼容；或复杂硬模式改变，它使在所述模式改变前的第一模式版本与所述模式改变后的第二模式版本不兼容。根据本方法，模式改变的执行划分为若干阶段(1, 2, 3, 4, 5)。根据一个实施例，模式改变划分为3个阶段，准备阶段(1)，扫描阶段(2)和结束阶段(3)。根据另一实施例，在扫描阶段(2)和结束阶段(3)之间可以执行另外两个阶段。另外的这两个阶段是评估阶段(4)和决定阶段(5)。



1. 执行数据库中复杂软模式改变或复杂硬模式改变的一种方法，其中在软模式改变中在模式改变前的第一模式版本与在模式改变后的第二模式版本兼容，而在硬模式改变中所述模式改变前的第一模式版本与所述模式改变后的第二模式版本不兼容，其特征在于划分模式改变过程为若干阶段：

- 第一阶段，准备阶段，包括在允许根据所述第一模式版本的事务处理得以继续的同时，准备要开始的根据所述第二模式版本的事务处理，该第一阶段是简单模式改变；

- 第二阶段，扫描阶段，包括转换根据所述第一模式版本的数据为根据所述第二模式版本的数据；

- 第三阶段，结束阶段，包括清除在模式改变完成后不应存在的那些单元，其为简单模式改变，并且

还在于，所述准备阶段包括应用不同的触发器、键字、外来键字和/或索引，以使所述第二模式版本可以根据影响所述第一模式版本的事务处理来更新，以及，在所述复杂模式改变是复杂软模式改变的场合，应用不同的触发器、键字、外来键字和/或索引，以使所述第一模式版本可以根据影响所述第二模式版本的事务处理来更新，或者其中，所述复杂模式改变是复杂硬模式改变，应用不同的触发器、键字、外来键字和/或索引，以使所述第一模式版本可以根据影响所述第二模式版本的事务处理来更新。

2. 根据权利要求1的方法，其中所述复杂模式改变是复杂软模式改变，其特征在于，允许根据所述第二模式版本的普通事务处理在完成所述数据转换后立即开始，即使根据所述第一模式版本的事务处理仍应继续。

3. 根据权利要求1的方法，其中，所述复杂模式改变是复杂硬模式改变，其特征在于，允许根据所述第二模式版本的普通事务处理在根据所述第一模式版本的所有事务处理已经完成后立

即开始。

4. 根据权利要求1的方法，其特征在于，所述结束阶段在根据所述第一模式版本的所有事务处理完成后开始。

5. 根据权利要求1的方法，其特征在于，所述复杂模式改变包括另外两个阶段；所述这两个阶段位于所述扫描阶段和所述结束阶段之间，其中

- 第一中间阶段，评估阶段，包括执行根据所述第二模式版本的某些评估事务处理，从而可以评估所述第二模式版本的功能；以及其中

- 第二中间阶段，决定阶段，包括决定模式改变应完成还是中断，这一决定根据所述第三评估阶段的结果做出。

6. 根据权利要求5的方法，其中所述复杂模式改变是复杂模式改变，其特征在于，评估事务处理是专门配置的控制事务处理。

7. 根据权利要求5的方法，其中所述复杂模式改变是复杂软模式改变，其特征在于，评估事务处理是根据所述第二模式版本的普通事务处理。

8. 根据权利要求5的方法，其中所述复杂模式改变是复杂软模式改变，其特征在于，允许根据所述第二模式版本的普通事务处理在所述决定阶段做出要完成所述模式改变的决定后立即开始，即使根据所述第一模式版本的事务处理仍在继续。

9. 根据权利要求5的方法，其中所述复杂模式改变是复杂硬模式改变，其特征在于，评估事务处理是专门配置的控制事务处理，它们可以根据所述第二模式版本执行而对所述第一模式版本无有害影响。

10. 根据权利要求5的方法，其中所述复杂模式改变是复杂硬模式改变，其特征在于，允许根据所述第二模式版本的普通事务处理在所述评估阶段做出要完成所述模式改变的决定和在根据所述第一模式版本的所有事务处理已经完成后立即开始。

11. 根据权利要求5的方法，其特征在于，所述结束阶段根据在所述评估阶段做出的决定、且如果所述决定是要完成所述模式

改变，在根据所述第一模式版本的所有事务处理已经完成后开始。

12. 根据权利要求 1 的方法，其特征在于，当所述模式改变被执行时，所述结束阶段包括清除不由所述第二模式版本使用的任何触发器、键字、外来键字、索引和/或其它单元；以及当所述模式改变要被放弃时，所述结束阶段包括清除不由所述第一模式版本使用的任何触发器、键字、外来键字、索引和/或其它单元。

13. 根据权利要求 5 的方法，其与软件改变相结合，所述软件改变中当前软件要由新软件替换，或者无较早对应软件的新软件将在所述数据库中或由所述数据库的用户实现，其特征在于，所述模式改变与所述软件改变同步；由所述新软件产生的事务处理组成所述评估事务处理；在所述评估阶段中的评估包括评估所述第二模式版本、评估所述新软件和评估它们的相互协同作用；和根据从所述评估阶段得到的结果做出是否要实现所述新软件和执行所述模式改变的决定。

14. 根据权利要求 1 的方法，其特征在于，所述模式改变包括存储信息到 SAGA 表中；所述信息既包括 UNDO 信息，也包括 REDO 信息，其中所述 UNDO 信息包括在所述准备阶段或所述扫描阶段、也可能在评估阶段或者决定阶段由于某种原因中断或放弃模式改变的事件下复位所述第一模式版本所需要的信息，以及其中所述 REDO 信息包括由于某种原因在所述结束阶段中断或放弃该模式改变的事件下为完成所述模式改变所需要的信息。

15. 根据权利要求 14 的方法，其特征在于，所述准备阶段包括存储所述 UNDO 信息，而所述扫描阶段包括存储所述 REDO 信息。

16. 根据权利要求 5 的方法，其特征在于，所述模式改变包括存储信息到 SAGA 表中；所述信息既包括 UNDO 信息，也包括 REDO 信息，其中所述 UNDO 信息包括在所述准备阶段或所述扫描阶段、也可能在评估阶段或者决定阶段由于某种原因中断或放弃模式改变的事件下复位所述第一模式版本所需要的信息，以及其中所述 REDO 信息包括由于某种原因在所述结束阶段中断或放弃该模式改变的事件下为完成所述模式改变所需要的信息。

17. 根据权利要求 16 的方法，其特征在于，所述准备阶段包括存储所述 UNDO 信息，而所述评估阶段包括存储所述 REDO 信息。

数据库的模式改变方法

技术领域

本发明涉及执行数据库内模式改变的方法。

本发明特别涉及执行复杂的模式改变的方法，而不管这一改变是软改变还是硬改变。然而，应该理解，本发明的方法同样可以应用于实现简单模式改变。

背景技术

很久以来就知道使用数据库中的表，它们以不同方式由不同事务处理连续地更新或改变。这种改变称为模式改变。还有一些不导致模式改变的事务处理。

模式指的是数据库中表的主要配置。该种配置可以包括：

- 表的格式，亦即表的名称，属性，分段和不同的安全状况；
- 分配给不同表列的属性，亦即该属性可以存储的信息类型；
- 存在什么键字，亦即外部引用如何翻译为内部引用；
- 发现什么样的外来键字，亦即在各表之间不同连接的本质；
- 发现什么样的不同索引，亦即对表中不同属性发现什么样的

搜索路径；

- 对一个表连接有什么样的触发器，亦即启动，“触发”，另一事件的特殊事件，使得表中数据的变化触发另一表按照该变化和按照现有键字，或许是外来键字，更新。

然而，术语模式隐含比上述更多的内容，但是该术语的上述意义是将与本发明下述说明结合使用的某些意义。

在分布式数据库中，把不同的表分成多个不同的段。这样的段依次又包含该表同样部分的一些拷贝或复制。不同段及其各自的拷贝物理上在相互协同操作的计算机的一个系统的不同节点中找到。系统的可靠性通过散布属于一个段的各种拷贝而改善，因为这使信息在该系统中某个地方可以使用，即使一个或者多个系统节点崩溃。

因此，分布式数据库内的模式改变可以相对复杂，因为由模式改变影响的段的所有拷贝必须根据这一改变而更新。

数据库中模式改变基于不同条件，常常包括一些不同的参加者，

引起模式改变产生不同表和 / 或，当数据库是分布式数据库时，产生数据库中一或多个表的不同部分、段中的改变。

模式改变导致该数据库所有用户的条件的改变和这些用户之间的事务处理的条件的改变。还可以提及不同的模式版本，其中，模式改变导致用第二模式版本替换第一模式版本。

模式改变可以分为不同类型。首先，模式改变可以是“简单的”或“复杂的”，其次，可以是“软改变”或“硬改变”。这些改变特性彼此部分独立。复杂模式改变既可以是硬改变，也可以是软改变，而简单模式改变总是软改变。

简单模式改变只改变模式，而不改变受影响表内的数据。复杂的模式改变既改变模式，也改变一个或者多个表中的数据。

简而言之，第一和第二模式版本在软模式改变中互相兼容。然而，在硬模式改变的场合，第二模式版本与第一模式版本不兼容。

兼容性指的是，根据第一模式版本的事务处理通过映射或一个函数可以更新根据第二模式版本的表中的数据，而根据第二模式版本的事务处理通过映射或一个函数可以更新根据第一模式版本的表中的数据。

这样，前后模式版本彼此兼容的软模式改变允许同时根据两种模式版本执行事务处理。

在硬模式改变的场合，使可用来自第一模式版本的数据更新第二模式版本中的表或反过来执行的映射或函数是不可逆的，换句话说，第一模式版本中的表不能用第二模式版本中的数据更新，反之亦然。

这样，影响受硬模式改变影响的单元的所有开始的事务处理必需在一个模式改变可以开始前根据第一模式版本结束。仅当模式改变已经完成，事务处理才可以按照第二模式版本开始。

这样，在结束正在进行的事务处理和执行所关心的模式改变占用的时间期间不能开始根据第二模式版本的事务处理。这些事务处理必须等待根据第一模式版本的事务处理和要执行的模式改变结束，导致相当大的时间损失。

复杂的模式改变需要几个阶段，因为这些改变本质上既影响数据也影响模式。复杂的模式改变包括一些相互顺序的模式改变和事务处理。

下述出版物说明涉及执行模式改变方法的公知的技术，还公开了在该领域遇到的各种问题。

由 R. J. Peters 和 M. T. Oezsu 发表的“对象库中的动态模式发展的公理化”，第 11 届国际数据工程会议，1995 年 3 月，156-164 页。

由 B. S. Lerner 和 A. N. Habermann 发表的“超越模式发展到数据库重组”，论文集，ACM OOPSLA/ECOOP '90，67-76 页，渥太华，加拿大，1990 年 10 月。

由 B. S. Lerner 发表的“为复杂类型改变的类型发展支持”，技术报告 UM-CS-94-71，马萨诸塞州立大学；Amherst，1994 年 10 月 31 日。

S. M. Clemen 所著“模式发展和集成，分布式和并行数据库”，第 2 卷，1994 年 1 月 1 日，ISSN0926-8782。

还应该提到，在本领域中公知所谓的 SAGA 事务处理与数据库的结合使用。在这一方面，有可能使用后继第二事务处理执行第一事务处理，第二事务处理将取消由第一事务处理引起的改变。

至于在这一特定领域中较早的技术观点，可以提到，当改变一个计算机系统软件时，公知与旧软件并行安装新软件。

随着这一安装，在旧软件清除前可以评估或测试新软件。如果测试表明新软件不能满意地工作，则可以将其清除，再次取旧软件使用。如果测试表明新软件能满意地工作，则取用新软件，清除旧软件。US-A 5410703 给出这种方法的一个例子。

当考虑上述现有技术状态时，可以看出，一个技术问题是为执行数据库内复杂模式改变提供一种方法。

另一技术问题是提供这样一种方法，使用这种方法可以执行复杂的模式改变，诸如复杂的软模式改变或复杂的硬模式改变，而且该方法能减少发生在复杂硬模式改变中由系统停止同时等待根据第一模式版本的事务处理的执行完成和等待实际模式改变实现引起的损失。

另一技术问题是使复杂模式改变分成不同阶段，其中，各阶段自身可以包括简单的模式改变或表更新。

另一技术问题在于提供一种方法，它考虑该模式改变是软改变还

是硬改变，但它在这两种场合都工作。

另一技术问题是提供一种方法，它允许在清除第一模式版本前相对于一个希望的功能评估第二模式版本。

另一技术问题是提供一种方法，它能使新实现的软件与结合新软件的实现需要的模式改变结合起来评估，此时所述评估可以用作决定新软件的实现及由此引起的模式改变是否应实行的基础。

另一技术问题是提供一种方法，其中，当发现第二模式版本不满足需要时可以返回到第一模式版本。

另一技术问题是提供处理复杂模式改变中的中断的一种方法，其中，模式改变可以继续或可以返回到原来的模式版本而不管该中断。

发明内容

为了解决上述一个或者多个技术问题，本发明提供执行数据库中复杂软模式改变或复杂硬模式改变的一种方法，其中在软模式改变中在模式改变前的第一模式版本与在模式改变后的第二模式版本兼容，而在硬模式改变中所述模式改变前的第一模式版本与所述模式改变后的第二模式版本不兼容，其特征在于划分模式改变过程为若干阶段：

- 第一阶段，准备阶段，包括在允许根据所述第一模式版本的事务处理得以继续的同时，准备要开始的根据所述第二模式版本的事务处理，该第一阶段是简单模式改变；

- 第二阶段，扫描阶段，包括转换根据所述第一模式版本的数据为根据所述第二模式版本的数据；

- 第三阶段，结束阶段，包括清除在模式改变完成后不应存在的那些单元，其为简单模式改变，并且

还在于，所述准备阶段包括应用不同的触发器、键字、外来键字和/或索引，以使所述第二模式版本可以根据影响所述第一模式版本的事务处理来更新，以及，在所述复杂模式改变是复杂软模式改变的场合，应用不同的触发器、键字、外来键字和/或索引，以使所述第一模式版本可以根据影响所述第二模式版本的事务处理来更新，或者其中，所述复杂模式改变是复杂硬模式改变，应用不同的触发器、键字、外来键字和/或索引，以使所述第一模式版本可以根据影响所述第二模式版本的事务处理来更新。

整个模式改变由第三阶段结束，它包括一个结束阶段，该阶段包

括清除在该模式改变完成后不应再保留的那些单元。

为能够评估第二模式版本的功能和返回第一模式版本的目的，本发明建议，可以在扫描阶段和结束阶段之间使用两个中间阶段。

第一中间阶段包括一个评估阶段，其中，按照第二模式版本执行某些评估事务处理，以之评估所述第二模式版本的功能。

跟随该阶段的是第二中间阶段，它包括决定阶段，在其中决定所述模式改变是完成还是放弃，这一决定根据评估阶段的结果作出。

当复杂模式改变是软模式改变且目的是使第一和第二模式版本并行操作时，建议准备阶段包括应用不同的触发器、键字、外来键字和/或索引，以之使第二模式版本可以根据影响第一模式版本的事务处理更新，以及应用触发器、键字、外来键字和/或索引，以之使第一模式版本可以根据影响第二模式版本的事务处理更新。

在这一场合，评估事务处理可以是根据第二模式版本的普通事务处理。

如果该改变是软模式改变的话，可以允许根据第二模式版本的普通事务处理在数据传输完成后立即开始，或当中间阶段被选择时，立即决定完成在决定阶段的模式改变，即使根据第一模式版本的事务处理仍在进行。

当复杂模式改变是复杂硬模式改变时，本发明建议，准备阶段包括应用不同的触发器、键字、外来键字和/或索引，以之使第二模式版本可以根据影响第一模式版本的事务处理更新，或者应用触发器、键字、外来键字和/或索引，以之使第一模式版本可以根据影响第二模式版本的事务处理更新，这取决于根据在两种模式版本之间的主要兼容性可以传输数据的方向。

由于第一和第二模式版本彼此不兼容，因此本发明建议，评估事务处理包括特别设计的控制事务处理，它们可以根据第二模式版本执行，而对第一模式版本没有有害影响。

在这一特定场合，允许在根据第一模式版本的所有事务处理完成后立即开始根据第二模式版本的普通事务处理，而且如果选择中间阶段的话，立即在决定阶段作出完成该模式改变的决定。

在根据第一模式版本的所有事务处理完成后，并且如果选择中间阶段的话，根据在决定阶段关于完成该模式改变得出的决定开始结束

阶段，而不管该模式改变是硬改变或软改变。

结束阶段还包括清除模式改变后主导模式版本不使用的所有触发器、键字、外来键字、索引和/或其它单元。

为能够对将实现的软件进行评估，本发明建议，在实现新软件时需要的模式改变与该实现并行进行。

根据本发明的一个实施例，当检验新软件时使用由该新软件在模式改变的评估阶段作为评估事务处理产生的事务处理。

在评估阶段获得的结果用作决定新软件的实现是否进行到底的基础，还用作决定该模式改变是否应完成。

为能在中断事件下恢复第一模式版本或者完成模式改变，本发明建议，模式改变包括在所谓的 SAGA 表中存储信息。

根据本发明，这一信息包括 UNDO 信息和 REDO 信息两种，UNDO 信息包括在准备、扫描、评估或决定阶段由于任何原因放弃模式改变的场合下重新建立第一模式版本需要的信息，而 REDO 信息包括在结束阶段由于某种原因在放弃模式改变的场合下进行到底或完成该模式改变需要的信息。

根据本发明的一个实施例，准备阶段包括存储必要的 UNDO 信息，而扫描阶段或，当选择中间阶段的话，决定阶段包括存储必要的 REDO 信息。

主要由本发明的方法提供的优点在于，执行复杂的模式改变的可能性，而不管这些改变是软改变还是硬改变，其中，模式改变和在某些场合下根据第二模式版本的事务处理可以在根据第一模式版本的所有事务处理结束前开始。实际上，在根据第一模式版本的事务处理结束前除结束阶段外可以执行完全的模式改变，相对于必须等待根据第一模式版本的所有事务处理在模式改变可以开始之前结束的场所，它提供了明显的时间增益。

本发明提供的另一优点是，可以在第一模式版本丢失前评估第二模式版本。本发明还能够使在对第二模式版本的评估表明该版本不能满意工作时返回到第一模式版本。

本发明还允许模式改变被完成或放弃，和在该模式改变偶然中断的事件下，诸如在存储或处理受该模式改变影响的表的硬件崩溃的事件下，返回到第一模式版本。

附图说明

为更容易理解本发明和使其另外的特征更为明显，参考本发明的示范性实施例以及附图详细说明本发明，其中，

图 1 是说明本发明方法的流程图；

图 2 是对与软模式改变的一个例子关联的第一和第二模式版本的概要说明；

图 3 是根据图 2 的模式改变部分执行的概要说明，并表示对一个外来表的处理；

图 4 概要说明根据图 2 的模式改变的可能结果；

图 5 是对与硬模式改变的一个例子关联的第一和第二模式版本的概要说明；

图 6 是根据图 5 的模式改变的部分执行的概要说明；

图 7 是对根据图 5 的模式改变的可能结果的概要说明；

图 8 是说明组合的模式改变和软件改变的流程图；以及

图 9 概要说明根据本发明的 SAGA 表。

具体实施方式

参考图 1 的流程图，说明根据本发明的方法。

根据这一方法的模式改变引起第一模式版本改变为第二模式版本。如图 1 所示，通过把该改变分成一些阶段 1、2、3、4、5，执行该模式改变。

根据本发明的概念，复杂模式改变分成 3 个阶段 1、2、3。可以允许该方法包括两个另外的中间阶段 4、5，以使在清除第一模式版本前评估第二模式版本。

只使用 3 个基本阶段 1、2、3 在某些场合提供更快的方法，虽然会增加不确定性，因为只使用 3 个基本阶段不能使第二模式版本在取用前评估。另外，在某些硬模式改变的场合，可能很难使用两个中间阶段。

第一阶段 1 是准备阶段，其中产生能使根据第二模式版本的事务处理开始的条件，同时保持允许根据第一模式版本的事务处理继续的条件。

第二阶段是扫描阶段，它包括将根据第一模式版本的数据传输到根据第二模式版本的数据。

第三阶段是结束阶段，它包括清除在完成模式改变后不应留下的那些单元。

上述阶段 1、2、3 是 3 个基本阶段。

第一中间阶段 4 是评估阶段，它可在 6 选择，虽然它不是绝对必需。该阶段是两个中间阶段中的第一个，包括执行根据第二模式版本的某些评估事务处理，从而对所述第二模式版本的功能进行评估。

在模式改变中，第二模式版本可以评估到不同程度，取决于什么是可能的和希望的。

在某些软模式改变中，有可能执行总系统测试，它包括执行一些评估事务处理，从而评估不同用户的功能及其可能的需求。

有可能包括较少数目的特定事务处理的功能测试便已足够。

在某些硬模式改变的场合，可能只能执行简单的基本测试，或者全然不能执行测试，取决于在第二模式版本中可以执行的评估事务处理，而不在两个模式版本之间产生不一致的程度。

当在 6 选择评估阶段 4 时执行第二中间阶段 5，它为决定阶段，其

中决定该模式改变是完成还是放弃。该决定基于来自评估阶段 4 的结果。

当复杂的模式改变是软改变时，本发明建议，准备阶段 1 包括应用不同触发器、键字、外来键字和 / 或索引。

这些触发器和键字使第二模式版本能根据影响第一模式版本的事务处理更新。还应用触发器、键字、外来键字和 / 或索引，使第一模式版本能根据影响第二模式版本的事务处理更新。

这意味着，根据一种模式版本的事务处理将触发根据该事务处理和所应用的键字的另一模式版本的更新，而不管触发事务处理是根据第一模式版本还是根据第二模式版本。

根据只使用 3 个基本阶段的概念，可以允许根据第二模式版本的普通事务处理在扫描阶段 2 已经执行后立即开始，亦即当数据从第一模式版本已经传输到第二模式版本时，即使根据第一模式版本的事务处理可能仍然在进行。

当复杂的模式改变是硬模式改变时，本发明建议，准备阶段 1 包括应用不同的触发器、键字、外来键字和 / 或索引，它使第二模式版本能根据影响第一模式版本的事务处理更新，或者应用触发器、键字、外来键字和 / 或索引，使第一模式版本可以根据影响第二模式版本的事务处理更新。

在这一场合，键字只能在一个方向作用，因为在两个模式版本之间缺少兼容性。

在硬模式改变的场合，允许根据第二模式版本的普通事务处理在根据第一模式版本的所有事务处理一旦完成后开始。

当根据第一模式版本的所有事务处理已经完成时开始结束阶段，而不管该模式改变是硬改变还是软改变。

当在软模式改变中使用两个中间阶段 4、5 时，本发明建议，评估事务处理可以包括对根据第二模式版本的普通事务处理的选择。

然后允许根据第二模式版本的普通事务处理在决定阶段 5 已作出要完成模式改变的决定时立即开始，即使根据第一模式版本的事务处理仍在进行。

当在硬模式改变中使用两个中间阶段 4、5 时，本发明建议，评估事务处理是特别设计的控制事务处理，其设计为根据第二模式版本执行，不对第一模式版本或数据库产生有害影响。

无有害影响指得是，在两个模式版本之间不出现不一致，意味着必须以这样的方式配置控制事务处理，使得尽管执行所述控制事务处理，但在这两个模式版本内的信息相等或一致。

然后允许根据第二模式版本的普通事务处理在评估阶段 5 作出继续该模式改变的决定和根据第一模式版本的所有事务处理完成时立即开始。

本发明还建议，当使用两个中间阶段 4、5 时，结束阶段 3 根据在决定阶段 5 作出的决定开始，而且如果该决定是完成模式改变，则在根据第一模式版本的所有事务处理已经完成后执行这一改变。

不管是否使用这两个中间阶段，也不管在决定阶段可能作出的任何决定，当模式改变完成后，清除第二模式版本不用的任何触发器、键字、外来键字、索引和/或其它单元，而在决定阶段包括当模式改变被中断或者放弃时清除第一模式版本不用的任何触发器、键字、外来键字、索引和/或其它单元。

为使本发明更容易理解，现在说明模式改变和这些模式改变可以如何分成根据上述的不同阶段的一些具体例子。

第一个例子通过图 2 说明，它包括竖直划分具有 10 个属性的表 A1 为分别具有 6 个和 4 个属性的两个表 B1、B2，划分前组成第一模式版本 A，划分后组成第二模式版本 B，其中旧表 A1 中的头 6 个属性形成第一新表 B1，旧表 A1 中后 4 个属性形成第二新表 B2。这是一个复杂软模式改变。

上述改变可以以不同方式执行，下面的说明仅涉及这样的模式改变可以如何根据本发明执行的一个例子。

准备阶段 1: 如图 3 所示，该准备阶段包括导致产生具有 4 个属性的完全新的空表 B2 的模式改变。

具有 10 个属性的旧表 A1 保持不动。这是一个软模式改变。

从旧表 A1 向新建立的表 B2 传输数据需要的触发器和键字 AB1、AB2、AB3、...、ABn 在旧表 A1 中建立，它们是通过相对于根据第一模式版本 A 的旧表的事务处理更新新建立的表 B2 的另一属性 AB 建立的。

在实践中，这一另外的属性 AB 可以包括几个不同的属性，虽然它在图中为简单起见表示为单一属性。产生这些属性的方式对熟悉本技术领域的人十分明显。对在下面说明的以相同方式产生的其它附加属性同

样适用。

通过新建立的表 B2 中的附加属性 BA 也产生相对于根据第二模式版本 B 的新表 B2 的事务处理更新旧表 A1 所需要的触发器和键字 BA1、BA2、BA3、...、BA_n。

旧表 A1 中的前 6 个属性也形成新表 B1 中的 6 个属性。这意味着对于在旧表 A1 的前 6 个属性和第一新表 B1 之间的更新不需触发器和键字，因为根据第二模式版本 B 的将要影响第一新表 B1 的事务处理将直接作用于旧表 A1 中的前 6 个属性，以及因为根据第一模式版本 A 的将要影响旧表 A1 中的前 6 个属性的事务处理将直接作用于第一新表 B1。

给在现有表 A1 中的每一行分配一个检验标记 A1f，它设定为“0”，并在该行中根据第一模式版本 A 的信息传输到第二模式版本 B 时改变为“1”。

该检验标记也可以在从各行传输信息到第二模式版本 B 之前通过设定在属性 AB 中的键字 AB1、AB2、AB3、...、AB_n 为“空”实现，之后，结合传输各行，设定各键字 AB1、AB2、AB3、...、AB_n 指向表 B2 中的正确行。

这样，一特定行的具有值“空”的来自属性 AB 的指针表示，从旧表 A1 到新建立的表 B2 的预定给新建立的表 B2 的相应行的信息尚未传输，以及在新建立的表 B2 中的相应行在改变旧表 A1 中的该行时不应更新。

扫描阶段 2: 扫描旧表 A1 中的后 4 个属性，并根据应用的键字 AB1、AB2、AB3、...、AB_n 传输相关信息到第二新表 B2，并在使用时把检验标记 A1f 从“0”设定为“1”。

第二新表 B2 根据按照第一模式版本仍在进行的并且涉及旧表 A1 中的后 4 个属性的事务处理继续更新。当扫描和信息传输完成时，可以允许根据第二模式版本 B 的事务处理开始，而与尚未完成的根据第一模式版本 A 的事务处理并行，此时，涉及 4 个属性的事务处理对第二新表 B2 执行，而涉及新表 B1 中的 6 个属性的事务处理对旧表 A1 中的前 6 个属性执行。

当允许根据第二模式版本 B 的事务处理开始时，旧表 A1 通过在新建立的表 B2 中的属性 BA 中的键字 BA1、BA2、BA3、...、BA_n 相对于影响新建立的表 B2 的事务处理更新。

这样，在两个模式版本 A、B 之间获得了一致，可以获得正确的数据来保证旧表 A1 将为涉及旧表 A1 中的后 4 个属性且可能仍未完成的根据第一模式版本的事务处理包含正确数据。

评估阶段 4: 允许有限数目的根据第二模式版本的事务处理开始，以便确认第二模式版本是否正确工作。

决定阶段 5: 在第二模式版本正确工作时做出完成该模式改变或将其进行到底的决定。不正确的工作产生放弃该模式改变的决定。

当做出完成模式改变的决定时，允许根据第二模式版本 B 的普通事务处理在剩余的根据第一模式版本 A 的事务处理完成的同一时间发生。

当做出放弃模式改变的决定时，完成正在进行的根据第一模式版本 A 的事务处理，而不开始根据第二模式版本 B 的任何普通事务处理。

结束阶段 3: 当在决定阶段 5 做出完成模式改变的决定时，等待结束所有正在进行的根据第一模式版本 A 的事务处理。旧表 A1 中不在第二模式版本 B 中使用的部分，亦即后 4 个属性，根据图 4 去除。

如果在决定阶段 5 做出放弃该模式改变的决定，则清除新建立的表 B2，保持旧表 A1 不动。不需等待所有正在进行的根据第一模式版本 A 的事务处理结束执行这一点。

不管在决定阶段 5 所做出的任何决定，清除为更新在涉及不同表的事务处理中的各自表中所建立的另外属性 AB、BA 中的所有检验标志 A1f、触发器和键字。

这样，把竖直划分表的复杂软模式改变分为 5 个阶段，其中第一和最后阶段是简单模式改变，第二阶段仅包括信息传输，而第三和第四阶段检验第二模式版本，它允许根据第一和第二模式版本 A、B 两者的事务处理被同时处理。

第二个例子用在一个表中的属性改变来说明，它可以是一个复杂的硬模式改变。在图示例子中，实际上，这一改变是一个复杂的硬模式改变。

图 5 表示一个表，它包括属性 A' 1 和另一属性 A' 2，前者指示不同人的小时工资，后者指示各人在 1 个月内工作的小时数。这是第一模式版本 A'。

要做的改变包括用指示各人月薪或月工资的属性 B' 替换这两个属性 A' 1、A' 2。这是第二模式版本 B'。

该例包括从第一模式版本 A' 向第二模式版本 B' 转换函数“f”。该函数“f”是“月薪” = “小时工资” * “小时数”。

在该例中，不存在传输第二模式版本 B' 中的信息亦即“月薪” B' 1 到第一模式版本 A'，亦即分别为“小时工资” A' 1 和“小时数” A' 2 的可能性，因为不知道“月薪” B' 1 的改变是由于“小时工资” A' 1 的改变引起还是“小时数” A' 2 的改变引起，或是由两者引起。

因此，这两个模式版本彼此不兼容，从而其模式改变是硬模式改变。

在该例中，图中只表示出所涉及的属性。然而应该理解，这些属性是或可能是包括若干属性的表的一部分。

准备阶段 1: 图 6 表示在现有表中如何产生属性 B' 1，该属性由根据第二模式版本 B' 的新属性“月薪”构成。产生触发器 A' B' 1，以之使在属性“小时工资” A' 1 和“小时数” A' 2 的任何行中的改变也导致在相应行中属性“月薪” B' 1 的更新。

扫描阶段 2: 该阶段包括扫描各行中在属性“小时工资” A' 1 和“小时数” A' 2 中的信息；之后，在属性“月薪” B' 1 中通过建立的函数“f”和触发器 A' B' 1 的媒介放置正确的信息。这在根据第一模式版本 A' 的所有事务处理完成前开始。

评估阶段 4: 开始根据第二模式版本 B' 的专用评估事务处理。这些事务处理一定不能对根据第一模式版本 A' 的信息有害，换句话说，它们一定不能产生数据库中的错误，诸如在这两个模式版本 A'、B' 之间的不一致。

一个这样的评估事务处理可以是在属性“月薪” B' 1 中插入打算根据第二模式版本 B' 被插入的当前的、正确的月薪。“月薪” B' 1 的值对于“小时工资” A' 1 和“小时数” A' 2 的相应值要正确，以便在两个模式版本 A'、B' 中的信息都正确。

从而可以看到，在扫描阶段中从第一模式版本 A' 向第二模式版本 B' 的信息传输包括多个评估事务处理。

该例通过评估事务处理只提供第二模式版本 B' 的有限测试功能。

决定阶段 5: 当确认正确的功能时做出完成该模式改变的决定。不正确的功能产生放弃该模式改变的决定。

当做出完成模式改变的决定时，允许根据第二模式版本 B' 的普通事务处理在根据第一模式版本 A 的所有事务处理一完成时即开始。

如果做出放弃模式改变的决定，继续正在进行的根据第一模式版本 A' 的事务处理，而不开始根据第二模式版本 B' 的任何事务处理。

结束阶段 3: 当在决定阶段 5 做出完成模式改变的决定时，等待结束所有正在进行的根据第一模式版本 A' 的事务处理。这样在事务处理继续根据第一模式版本 A' 进行时，同时执行所有前述过程，而不管总模式改变是硬模式改变。

当结束根据第一模式版本的事务处理时，开始根据第二模式版本 B' 的普通事务处理。从图 7 可明显看出，清除在第二模式版本 B' 中不用的属性“小时工资” A' 1 和“小时数” A' 2。

如果在决定阶段 5 做出放弃该模式改变的决定，则清除新建立的属性 B' 1。不需等待结束正在进行的根据第一模式版本 A' 的事务处理便执行这一点。

不管在决定阶段 5 所做出的任何决定，清除为更新在事务处理中影响属性“小时工资” A' 1 和“小时数” A' 2 的属性“月薪” B1 而建立的所有触发器 A' B' 1。

这样，把前述复杂硬模式改变分为 5 个阶段，其中第一和最后阶段是简单模式改变，第二阶段仅包括信息传输，所述阶段允许根据第一模式版本 A' 的事务处理和第二模式版本 B' 的提前实现同时执行。

上面，参考两个例子说明模式改变，其中只叙述了由该模式改变直接影响的表。然而，从图 3 可以看出，模式改变也可以影响除由该模式改变直接影响的表之外的表，所谓的外来表，由于它们与由该模式改变直接影响的一张或多张表以某种方式协同作用。这种协同作用赖以发生的键字称为外来键字。

在上述根据软模式改变且示于图 2、3 和 4 中的例子中，这一协同作用在于，在外来表 C1 中的一个元素的改变也导致根据第一模式版本的旧表 A1 的改变，其中在外来表 C1 中的改变还导致在第一新表 B1 和第二新表 B2 两者中的改变。

根据第一模式版本 A 提供的在属性 CA 中的外来键字 CA4 是连接外来键字 C1 与旧表 A1 的链路。

除上述触发器和键字外，在模式改变的准备阶段还产生能使外来表 C1 和第二新表 B2 相互协同作用的触发器和外来键字 CB4。外来表 C1 还与第一新表 B1 通过属于属性 CA 的外来键字协同作用。

为简单起见，只表示出各属性 CA 和 CB 中的第四外来键字。然而应该理解，这些属性包括为前述协同作用所需要的所有键字。

取决于该模式改变是否完成，在结束阶段清除不为改变后所用模式版本使用的外来键字。

关于如何处理外来表和外来键字的说明仅作为例子给出。十分明显，这一处理过程将随不同模式改变变化很大。还可以在诸如上述例子的同一例子中以不同方式执行。

因为模式改变还影响数据库的不同应用，同步数据库中的或者数据库用户的软件或新软件的实现的可能改变与模式改变是适宜的。

图 8 的流程图意在表示软件改变 Y 是如何与模式改变 X 并行执行的。

模式改变 X 包括从第一模式版本 A"到第二模式版本 B"的改变，并从准备阶段 X1 和扫描阶段 X2 开始。软件改变 Y 的初始阶段 Y1 在同一时间开始，该阶段包括足以评估这一改变的软件的实现。

然后开始评估阶段 XY1，并由新软件 Y 产生使用第二模式版本 B"的事务处理，组成评估阶段 XY1 中的评估事务处理。

在评估阶段 XY1 执行的评估包括第二模式版本 B"和新软件 Y 两者以及它们的相互协同作用的评估。

在决定阶段 XY2 根据在评估阶段 XY1 获得的结果决定新软件 Y 的实现是否要完成。

根据在决定阶段 XY2 做出的决定，在结束阶段 X3、Y2 要么完成模式改变 X 和软件改变 Y，要么终止它们。

本发明还提供一个优选实施例，其中，模式改变包括在所谓 SAGA 表中的信息的存储。

如图 9 所示，SAGA 表由一个数据库表 S1 组成，它包括能使一个模式改变完成或在分布式数据库中受一个模式改变影响的一个或多个节点崩溃的事件下放弃的信息。

根据本发明，SAGA 表 S1 还可以用于非分布式数据库，以提供完成模式改变或在该模式改变期间硬件故障的事件下复位数据库到第一模式版本的同样的可能性。

SAGA 表 S1 的目的是能使模式改变以和根据现有技术的 SAGA 事务处理的实现类似的方式实现模式改变。然而，与模式改变结合的实现较为复杂，由于模式改变包括一系列事务处理，它们可以导致数据和模式

两者不同的改变。

SAGA 表 S1 包含的信息由 UNDO 信息 S1U 和 REDO 信息 S1R 组成，其中，UNDO 信息 S1U 包括在准备阶段 1 或扫描阶段 2、也可能在评估阶段 4 或者决定阶段 5 由于某种原因中断该模式改变的事件下复位第一模式版本 A 所需要的信息，REDO 信息包括如果由于某种原因在结束阶段 3 中断该模式改变，而为完成该模式改变所需要的信息。

模式改变的中断可以是计划的中断，例如由于对第二模式版本的评估表示该版本不能满意工作而决定不完成该模式改变。

中断或者终止也可以是非计划的，它由例如在分布式数据库中受该模式改变影响的一个节点的故障引起，或者由在非分布式数据库中的硬件故障引起。

如果只使用 3 个基本阶段的话，本发明建议，准备阶段 1 包括存储 REDO 信息 S1R，扫描阶段 2 包括存储 UNDO 信息 S1U。

另一方面，如果还使用两个中间阶段的话，则建议准备阶段 1 包括存储 REDO 信息 S1R 和在评估阶段 5 包括存储 UNDO 信息 S1U。

根据本发明，SAGA 表 S1 还可以在结合安全拷贝更新任何再生拷贝或全再生数据库中使用。

应该理解，本发明不限制于前述和图示示范性实施例，可以在下述权利要求定义的发明概念的范围内进行修改。

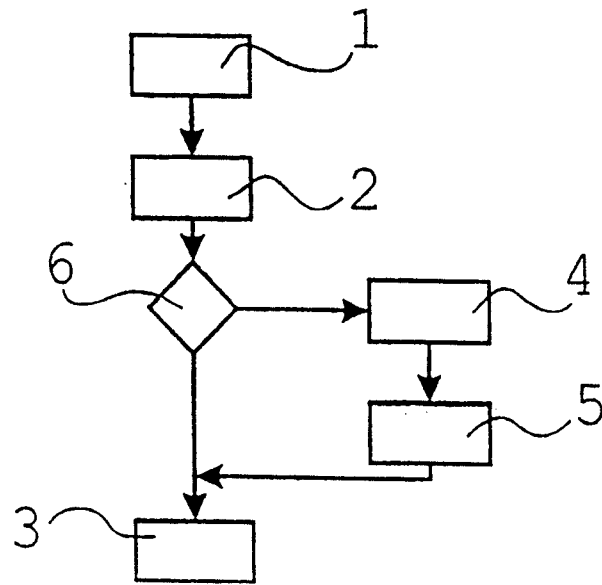


图 1

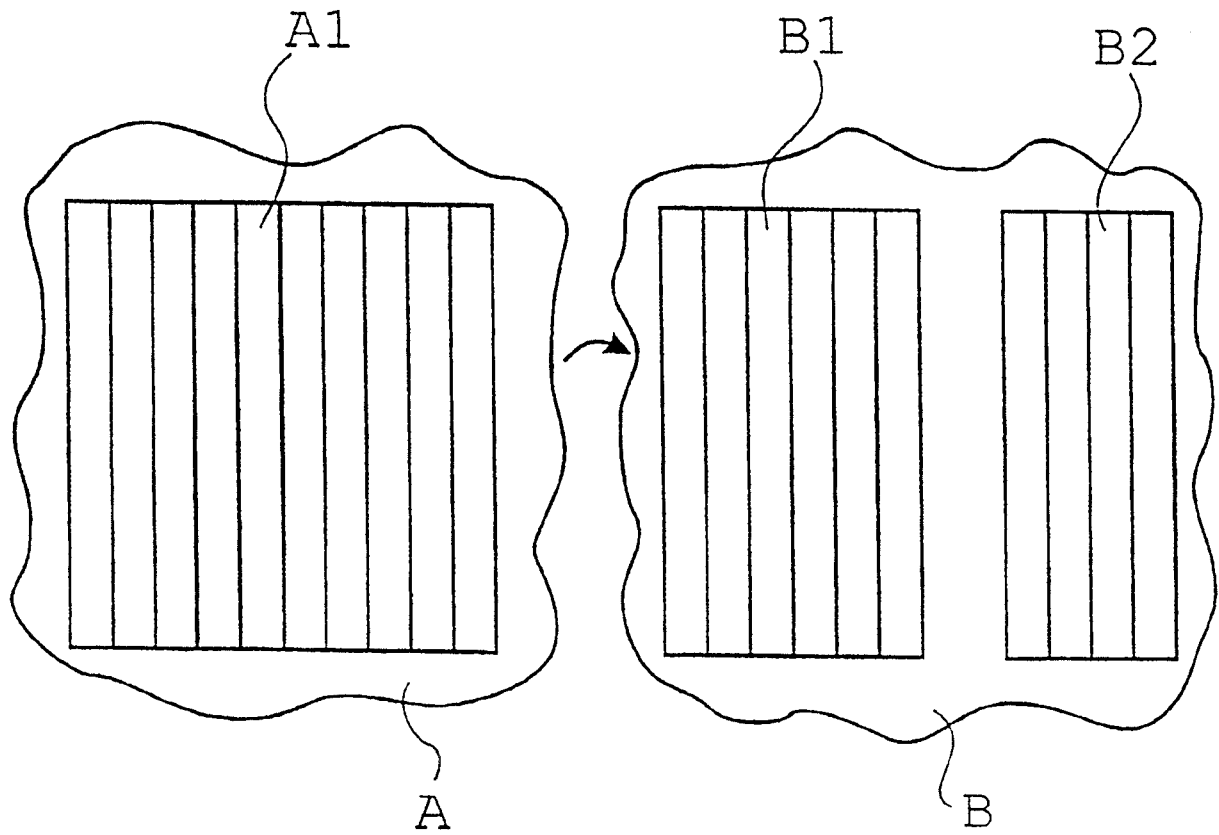


图 2

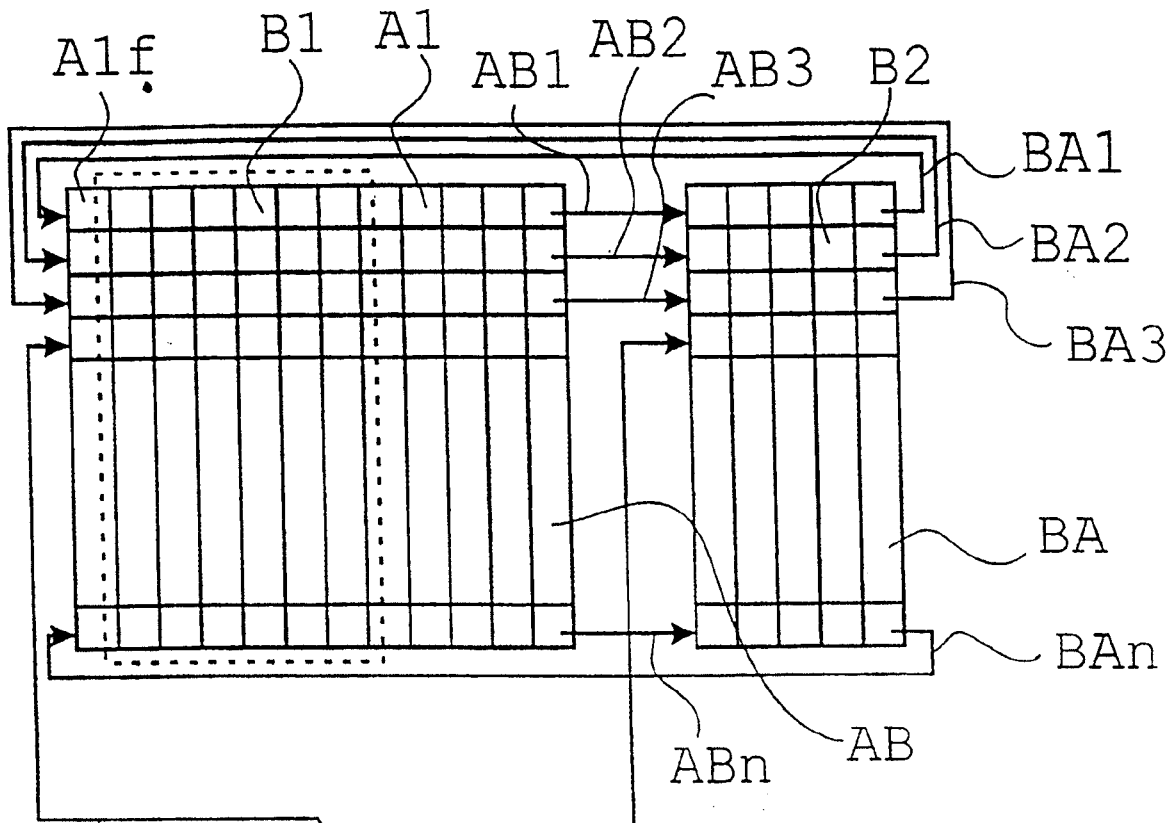


图 3

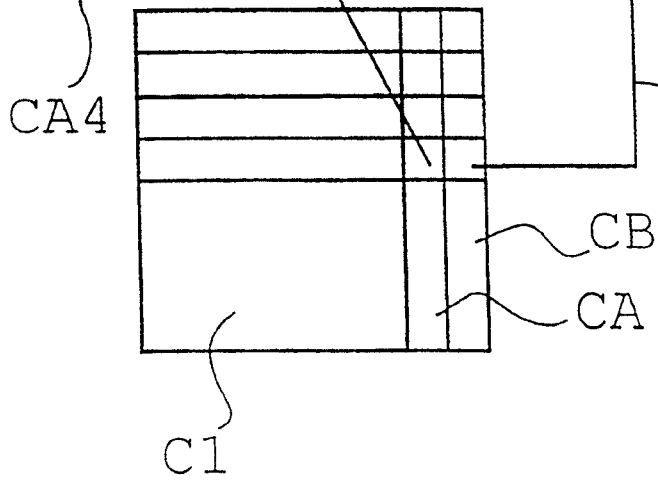
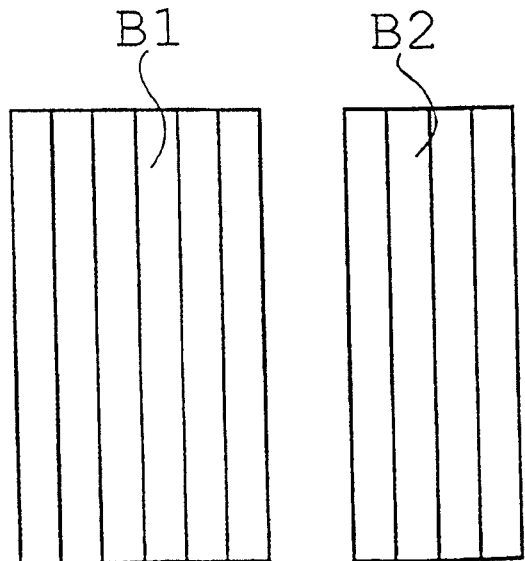


图 4



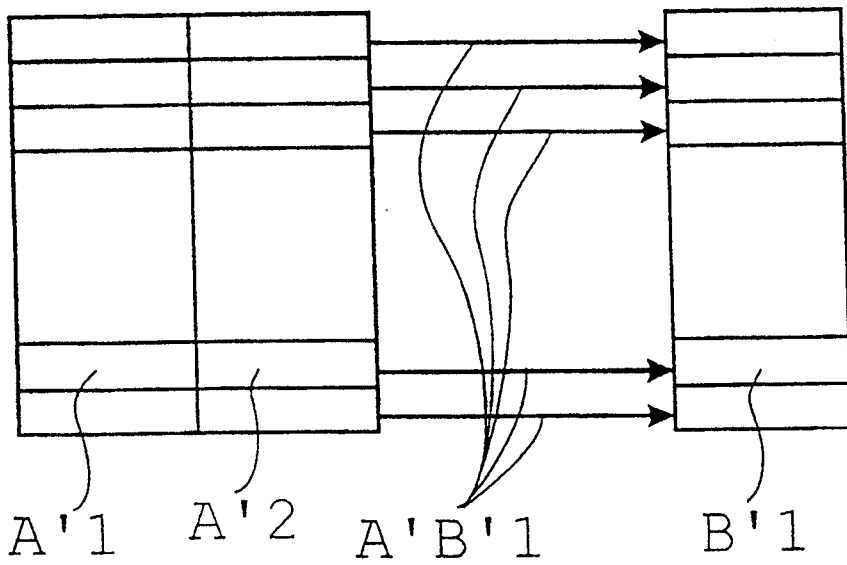
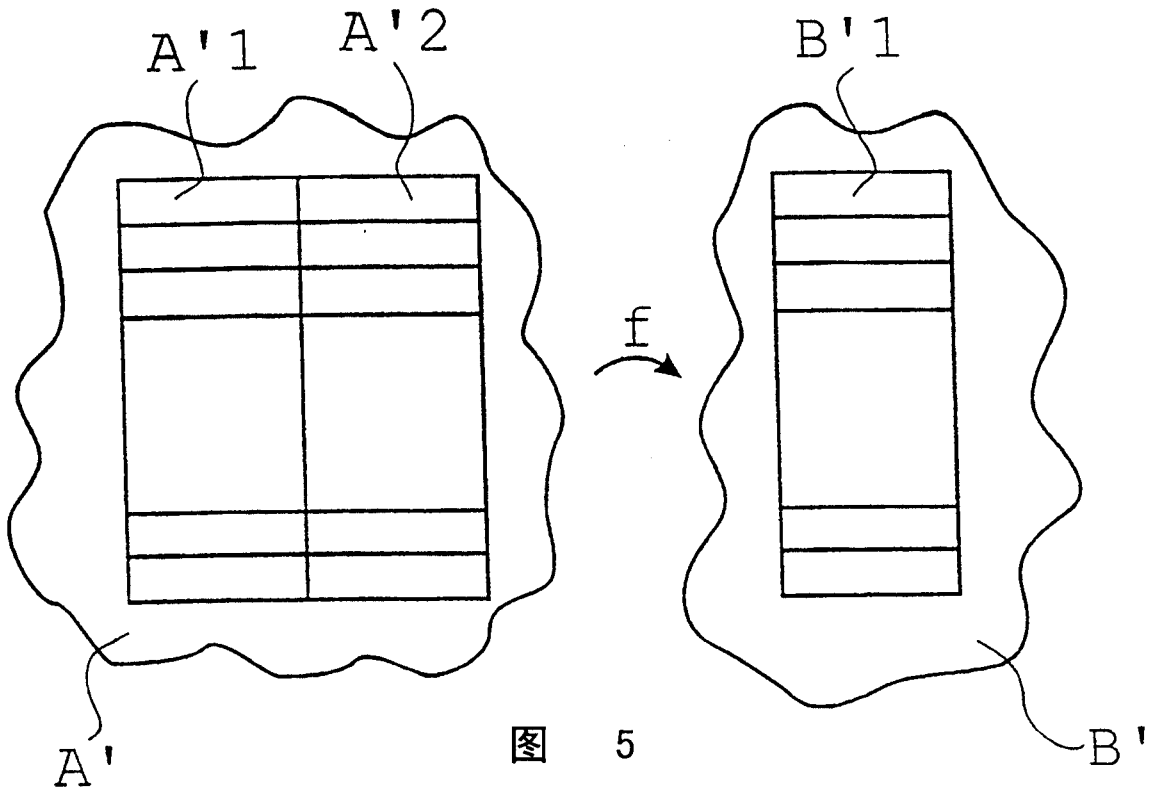


图 6

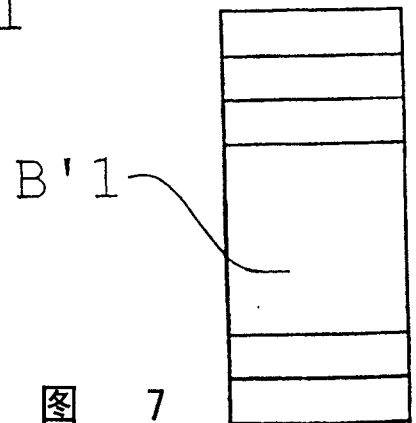


图 7

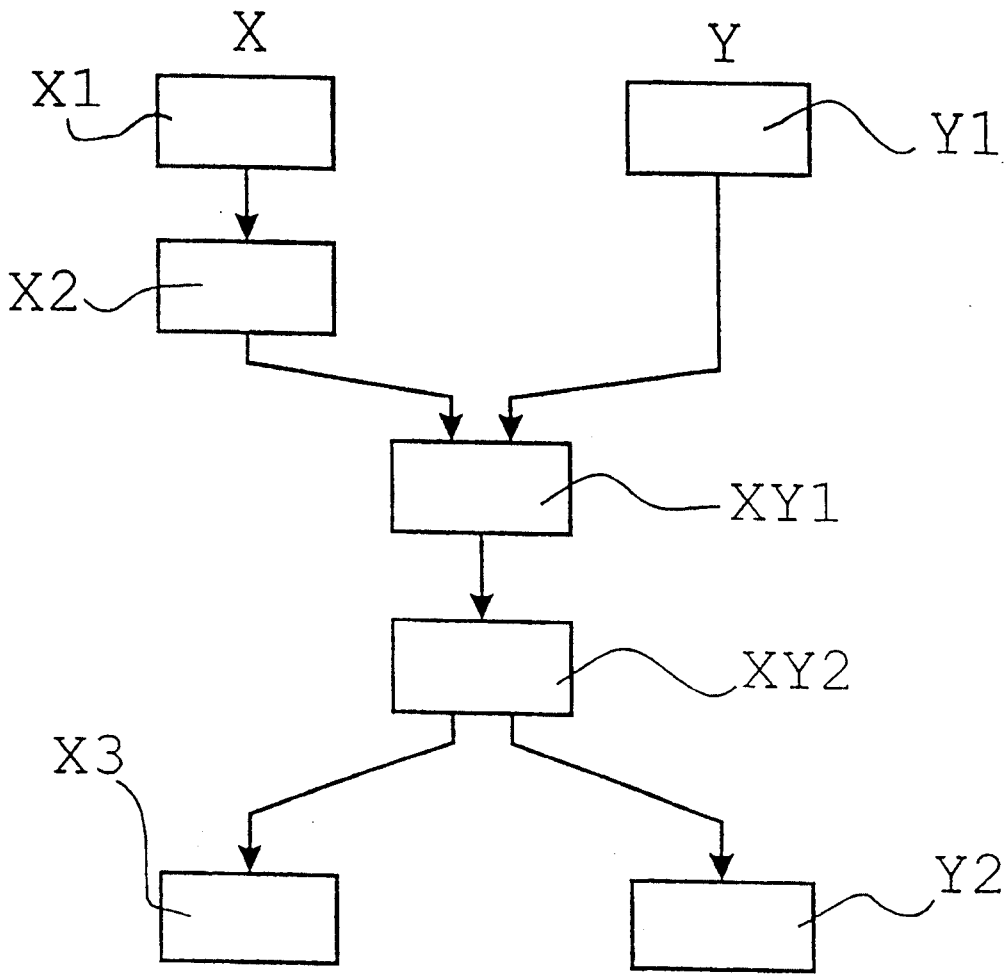


图 8

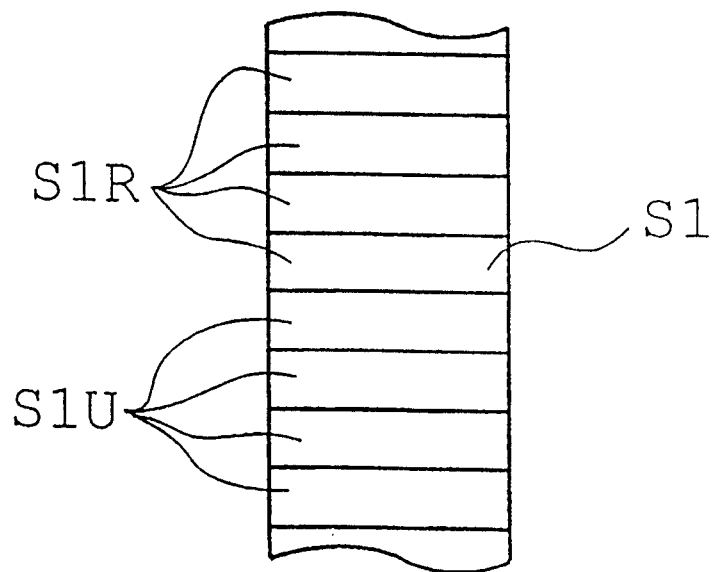


图 9