(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103745016 B
(45)授权公告日 2018.03.06

(21)申请号 201410045335.5

(22)申请日 2007.03.02

(65)同一申请的已公布的文献号
申请公布号 CN 103745016 A

(43)申请公布日 2014.04.23

(30)优先权数据
11/372,968 2006.03.10 US

(62)分案原申请数据
200780008489.5 2007.03.02

(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚

(72)发明人 萨史堪斯·查卓瑟卡兰

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 李晓芳

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

(56)对比文件
CN 1454349 A,2003.11.05,全文.
US 2005/0097149 A1,2005.05.05,全文.
US 2005/0289187 A1,2005.12.29,说明书[0026]-[0041],图3.
douglas terry 等.continuous queries over append-only databases.《ACM SIGMOD》.1992,第321-330页.

审查员 高沛沛

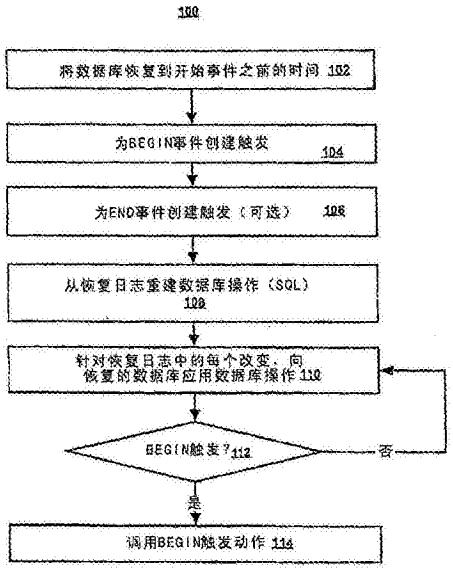
权利要求书3页 说明书12页 附图4页

(54)发明名称

利用恢复日志检测数据库事件

(57)摘要

提供了一种用于确定在数据库中何时发生事件的方法和装置。将数据库的至少一部分恢复到事件之前的点。将恢复日志转化成反映能够导致恢复日志中描述的改变的数据库操作(例如SQL)。创建用于基于对语句的执行来检测事件的机制。例如,创建数据库触发来检测事件。针对恢复的数据库执行数据库操作,以使该机制来检测该事件。



CN 103745016 B

1.一种由计算机实现的用于检测数据库中的事件的发生的方法，包括：

将所述数据库的至少一部分的拷贝恢复到时间性查询的开始事件的发生之前的点，其中所述时间性查询的结果是从所述数据库的多个先前的状态计算出的；

将描述对所述数据库的改变的重做日志记录转化成反映能够导致对所述数据库的所述改变的数据库操作的语句；

生成用于基于对所述语句的执行检测所述开始事件的第一数据库触发；

生成用于基于对所述语句的执行检测所述时间性查询的结束事件的第二数据库触发；

在将重做日志记录转化成所述语句以及生成第一数据库触发之后，使得所述第一数据库触发通过执行所述语句检测开始事件；

访问基于所述时间性查询的基本查询；

在所述第一数据库触发检测到所述开始事件后，执行以下步骤：

使得所述第二数据库触发通过执行所述语句来检测所述结束事件；以及

响应于所述第二数据库触发检测到所述结束事件，发出所述基本查询以确定对所述时间性查询的响应，其中，所述基本查询的结果提供对所述时间性查询的响应。

2.如权利要求1所述的方法，还包括生成对所述时间性查询作出响应的机制，其中所述机制确定在从所述开始事件开始的时间范围中对数据库对象做出的改变。

3.如权利要求2所述的方法，其中，所述用于对所述时间性查询作出响应的机制包括视图。

4.如权利要求3所述的方法，其中，所述视图是物化视图。

5.如权利要求3所述的方法，还包括将所述视图中的当前数据与所述视图中的先前数据相比较，以确定对所述数据库对象作出的改变。

6.如权利要求2所述的方法，其中，所述第一数据库触发激活所述对所述时间性查询作出响应的机制。

7.如权利要求1所述的方法，还包括：

确定在所述开始事件和所述结束事件之间不存在对数据库对象的更新或删除，来作为发出所述基本查询以确定对所述时间性查询的响应的条件。

8.如权利要求1所述的方法，还包括：

确定所述基本查询是单调的，来作为发出所述基本查询以确定对所述时间性查询的响应的条件。

9.一种由计算机实现的用于检测数据库中的事件的发生的方法，包括：

将所述数据库的至少一部分的拷贝恢复到时间性查询的开始事件的发生之前的点，其中所述时间性查询的结果是从所述数据库的多个先前的状态计算出的；

将描述对所述数据库的改变的重做日志记录转化成反映能够导致对所述数据库的所述改变的数据库操作的语句；

生成用于基于对所述语句的执行检测所述开始事件的第一数据库触发；

生成用于基于对所述语句的执行检测所述时间性查询的结束事件的第二数据库触发；

在将重做日志记录转化成所述语句以及生成第一数据库触发之后，使得所述第一数据库触发通过执行所述语句检测开始事件；

在检测到所述开始事件后，执行以下步骤：

基于所述时间性查询而激活视图；

执行一组语句，其中，该组语句反映能够导致在所述开始事件之后发生的对所述数据库的所述改变的数据库操作；

分析在该组语句被执行时对所述视图中的数据的改变，以确定对所述时间性查询的响应；以及

响应于所述第二数据库触发检测到所述结束事件，提供对所述时间性查询的所述响应。

10.一种用于检测数据库中的事件的发生的计算机系统，包括：

一个或多个处理器；和

存储器，存储有指令，所述指令当被所述一个或多个处理器执行时使得所述一个或多个处理器执行以下操作：

将所述数据库的至少一部分的拷贝恢复到时间性查询的开始事件发生之前的点，其中，所述时间性查询的结果是从所述数据库的多个先前的状态计算出的；

将描述对所述数据库的改变的重做日志记录转化成反映能够导致对所述数据库的所述改变的数据库操作的语句；

生成用于基于对所述语句的执行、针对恢复到所述开始事件发生之前的点的拷贝来检测所述开始事件的第一机制；

生成用于基于对所述语句的执行来检测所述时间性查询的结束事件的第二机制；

在将重做日志记录转化成所述语句以及生成所述第一机制之后，针对恢复到所述开始事件发生之前的点的拷贝来执行所述语句；

在执行所述语句时，导致所述第一机制检测所述开始事件；

访问基于所述时间性查询的基本查询；

在检测到所述开始事件后，执行以下步骤：

使得所述第二机制通过执行所述语句来检测所述结束事件：以及

响应于所述第二机制检测到所述结束事件，发出所述基本查询以计算对所述时间性查询的响应，其中，所述基本查询的结果提供对所述时间性查询的所述响应。

11.如权利要求10所述的计算机系统，其中，所述第一机制和第二机制包括数据库触发。

12.如权利要求10所述的计算机系统，其中，所述语句遵从SQL。

13.如权利要求10所述的计算机系统，其中，所述指令当被所述一个或多个处理器执行时进一步使得所述一个或多个处理器通过基于对所述语句的执行确定在时间性查询中指定的时间段中对数据库对象作出了什么改变，来对所述时间性查询作出响应，其中，所述时间性查询的结果是从所述数据库的多个先前的状态计算出的。

14.一种用于检测数据库中的事件的发生的计算机系统，包括：

一个或多个处理器；和

存储器，存储有指令，所述指令当被所述一个或多个处理器执行时使得所述一个或多个处理器执行以下操作：

将所述数据库的至少一部分的拷贝恢复到时间性查询的开始事件发生之前的点，其中，所述时间性查询的结果是从所述数据库的多个先前的状态计算出的；

将描述对所述数据库的改变的重做日志记录转化成反映能够导致对所述数据库的所述改变的数据库操作的语句；

生成用于基于对所述语句的执行、针对恢复到所述开始事件发生之前的点的拷贝来检测所述开始事件的第一机制；

生成用于基于对所述语句的执行来检测所述时间性查询的结束事件的第二机制；

在将重做日志记录转化成所述语句以及生成所述第一机制之后，针对恢复到所述开始事件发生之前的点的拷贝来执行所述语句；

在执行所述语句时，导致所述第一机制检测所述开始事件；以及

在检测到所述开始事件后，执行以下步骤：

基于所述时间性查询而激活视图；

执行一组语句，其中，该组语句反映能够导致在所述开始事件之后发生的对所述数据库的所述改变的数据库操作；

分析在该组语句被执行时对所述视图中的数据的改变，以确定对所述时间性查询的响应；以及

响应于所述第二机制检测到所述结束事件，提供对所述时间性查询的所述响应。

利用恢复日志检测数据库事件

[0001] 本分案申请是基于申请号为200780008489.5、申请日为2007年3月2日、发明名称为“利用恢复日志检测数据库事件”的中国专利申请(即,国际申请号为PCT/US2007/005351的PCT申请进入中国国家阶段后的申请)的分案申请。

技术领域

[0002] 本发明涉及数据库管理系统。具体地,本发明的实施例涉及利用数据库恢复日志(recovery log)来检测数据库中的事件。

背景技术

[0003] 数据库管理系统(DBMS)的一个重要功能是重建发生在数据库中的过去事件序列,以用于取证分析或者满足管理要求。构造这种事件可能涉及(DBMS)对时间性查询(temporal query)作出响应。时间性查询是这样一种查询,其中只能根据一个(或多个)先前的数据库状态来计算结果。考虑DBMS跟踪购买订单和用户角色的情形。用户的角色确定了用户在数据库系统中执行功能或动作(例如购买订单)的权力。在某个时刻,销售人员被错误地授予了批准购买订单的权力。例如,存储在数据库中的表中的销售人员角色被错误地改变成准予购买订单的权力。该销售人员于是开始批准购买订单,其中DBMS记录购买订单。在之后的某个时刻,审核员发现该销售人员不应当被授予该权力。审核员可以使该销售人员的角色被改变成不再允许购买订单批准,如果这种角色改变尚未被作出的话。审核员还需要知道该销售人员利用错误的权力批准的所有购买订单。在其当前状态下,数据库中可能没有该信息,因为购买订单的记录已经被删除或者记录中的授权信息已经被覆写,以反映具有合法权力的某人进行的购买订单批准。为了确定什么购买订单是由该销售人员批准的,审核员可能希望使用如下的时间性查询:

[0004] “请向我示出在John Doe是销售人员的时间期间他所批准的所有购买订单”。

[0005] 时间性查询的实际格式可能不同于该示例。示例性的时间性查询的开始时间是销售人员的角色变为“Salesman”时。时间性查询的结束时间是销售人员的角色不包括“Salesman”时。因此,DBMS应当返回在开始和结束时间之间该销售人员所批准的所有购买订单。

[0006] 存在其他的需要DBMS对时间性查询作出响应的情形。例如,操作员可能错误地将某个物品标记为“待售”。在这种情况下,希望确定DBMS所记录的所有以错误价格进行的对该物品的销售。

[0007] 确定对于这种时间性查询的答复提出了许多挑战。诸如一致读取、闪回查询等等之类的技术不是该问题的可行解决方案。一致读取或者闪回查询执行过去的特定时间的查询。但是,与时间性查询有关的开始和/或结束时间可能不是已知的。例如,利用销售人员的示例,特定销售人员的雇用日期和该问题被发现的时间可能是已知的。但是,错误地授予权力的时间可能不是已知的。另外,纠正该权力的时间可能不是已知的。

[0008] 另外,对时间性查询的响应应当提供开始时间和结束时间之间的所有所关注的结

果。例如，销售人员错误准予的所有购买订单都是所关注的。因此，诸如一致读取、闪回查询等等之类的技术不是该问题的可行解决方案。

[0009] 另外，对数据库进行到过去某个时间的时间点恢复(point-in-time recovery)不是可行的解决方案。如前所述，有关时间可能不是已知的。另外，时间点恢复不提供在某个时间范围中的信息。

[0010] 时间性查询问题的一种传统解决方案是应用编程者在应用内维护带时间戳的改变的详细踪迹。例如，编程者在内部转换每个更新或删除语句，以包括预处理或后处理步骤，该步骤记录数据的之前图像和作出改变的时间戳。这导致了额外的列和行被创建在数据库中，这些列和行可用于答复时间性查询。

[0011] 利用上述示例性时间性查询，可能存在角色表(“roles”)，该表可具有属性“role name”(角色名)，该属性可具有值“Salesman”，表明具有该角色的人具有购买订单权力。另外，订单表(“orders”)可具有属性“approver_name”(批准者姓名)，其值表明批准订单的人。为了答复时间性查询，应用编程者可以在两个表中都创建两个新的列“when”(时间)和“when_deleted”(删除时间)。编程者在“when”列中包括当前时间，作为更新或插入语句的一部分。另外，当行被删除时，编程者在“when_deleted”列中包括当前时间。(或者，编程者可以将行的删除改为新行的插入，该新行具有被设定为“DELETED”(已删除)的状态列。)对数据库的这些改变使得可以通过发出以下SQL语句来答复上述示例性时间性查询：

[0012] select*

[0013] from orders o,roles r

[0014] where o.approver_name=″John Doe″and r.name=″John Doe″and

[0015] r.role_name=″Salesman″and o.when between r.when and r.when_deleted

[0016] 但是，这种传统的技术具有多个缺陷。为了扩展数据库架构(schema)并正确地维护对数据库行的改变的踪迹，对应用编程者施加了额外的负担。性能受到损害，因为所有改变都被应用所审核。另外，数据库表的大小增加。大小增加的一个原因在于无法从表中删除时间戳信息，这意味着无法删除行。

[0017] 另一种传统的方法不是对应用作出上述改变，而是创建在数据库后端执行代码的数据库触发(database trigger)来实现类似的结果。Richard T.Snodgrass在“Developing Time Oriented Database Application in SQL”中描述了这种方法和其他有关问题。但是，与对应用编程者施加负担的技术一样，创建数据库触发也有着类似的问题。

[0018] 许多数据库应用对表施行独特的约束。例如，表被约束，以使得特定用户只能具有单个权限。但是，这使数据库不能在同一个表中具有针对特定用户的多个行。因此，在该示例中，同一个表无法被用于向John Doe赋予两个不同的角色。

[0019] 对于每个用户只允许一个角色的表的限制的一个解决方案是创建单独的历史表。通过将“历史性”的行移植到一个不同的分区，可以减轻这种损失。但是，架构设计者仍面临着为数据分区的负担。

[0020] 对于每个用户只允许一个角色的表的限制的另一解决方案是改变约束以添加时间戳，这是一个不同的过程。但是，应用该约束涉及应用许多规则，这些规则是通过对SQL语句进行重大修改来施行的。对SQL的修改会向应用或DBMS(如果这种功能被内置到DBMS中的话)添加大量开销和复杂性。

[0021] 即使很少发出时间性查询，也会导致与上述技术相关联的损失。例如，时间性查询可能只会作为审核期间的调查的一部分发生。另外，查询可能是针对数据库对象的小子集发出的，而损失却发生在对所有对象的改变上。

[0022] 因此，需要DBMS对时间性查询作出响应。需要不对应用编程者施加沉重负担或者不要求对数据库后端作出重大改变的技术。还需要不会对数据库的正常操作导致严重的性能损失的技术。还需要不要求大大增加数据库大小的技术。

[0023] 本部分中描述的方法是可以实行的方法，但不一定是先前已经被设想或实行过的方法。因此，除非另有表明，否则不应当假定本部分中描述的任何方法仅因为其被包括在本部分中就可以作为现有技术。

附图说明

[0024] 图1是示出根据本发明实施例用于检测在数据库中是否发生了事件的过程的流程图。

[0025] 图2是示出根据本发明实施例为时间性查询提供结果的过程的步骤的流程图。

[0026] 图3是示出根据本发明实施例利用物化视图(materialized view)对时间性查询提供结果的过程步骤的流程图。

[0027] 图4是示出本发明实施例可在其上实现的计算机系统的框图。

[0028] 在附图中以示例方式而非限制方式示出了本发明，附图中相似的标号指代类似的元件，其中：

具体实施方式

[0029] 在下面的描述中，出于说明目的，阐述了许多具体细节以帮助全面理解本发明。但是，很明显，没有这些具体细节也可实现本发明。在其他情况下，公知的结构和设备是以框图形式示出的，以避免不必要地模糊本发明。

[0030] 概述

[0031] 本发明的一个实施例是一种用于确定在数据库中何时发生了事件的方法。将数据库的至少一部分恢复到事件之前的时间点。将恢复日志转化成表示能够导致恢复日志中描述的改变的数据库操作的语句(例如SQL语句)。创建一种机制，用于基于对这些语句的执行来检测事件。例如，创建数据库触发以检测事件。针对经恢复的数据库执行语句，以使该机制来检测事件。

[0032] 本发明的另一个实施例是一种用于对时间性查询作出响应的方法。用户提供时间性查询，该时间性查询具有开始事件、结束事件和与开始和结束事件之间的时间有关的查询。可以通过与前一段中论述的实施例相类似的方式来检测开始事件。生成触发事件以检测结束事件，并且生成机制以确定查询块的结果。例如，生成物化视图，并通过执行表示基于恢复日志的数据库操作(例如SQL)的语句来填充该物化视图。基于对物化视图中的信息的分析来答复时间性查询。

[0033] 示例性时间性查询

[0034] 在一个实施例中，为对DBMS的时间性查询提供响应。出于说明目的，使用以下示例性时间性查询。用户可能希望知道对以下时间性查询的答复：

[0035] “请向我示出在John Doe是销售人员的时间期间他所批准的所有购买订单”。

[0036] 时间性查询的实际格式可以基于容易转换成数据库操作的格式。例如，时间性查询可以像以下示例那样基于SQL。一般地，时间性查询的格式是“BEGIN QUERY END”。BEGIN和END块指定开始和结束点。一般地，BEGIN和END块可以是事件指定。换言之，不指定时间。下面论述特殊情况，包括指定开始或结束时间的示例。在该示例中，开始事件是在数据库表中更新或插入一行，该行向用户“John Doe”赋予角色“Salesman”。因此，以下BEGIN块适合于描述该事件。“开始事件”被定义为当BEGIN块中的条件取值为“TRUE”(真)时。

[0037] BEGIN is AFTER INSERT OR UPDATE OF roles

[0038] WHERE roles.name=″John Doe″and roles.role_name=″Salesman″

[0039] 继续该示例，END块反映了这样一个数据库事件，该数据库事件描述John Doe何时不再具有销售人员的角色。以下的示例性END块指定了更新或删除何时从用户“John Doe”中删除角色“Salesman”。“结束事件”被定义为END块中的条件取值为“TURE”时。

[0040] END is AFTER DELETE OR UPDATE OF roles

[0041] WHERE roles.name=″John Doe″and roles.role_name=″Salesman″

[0042] 继续该示例，以下的示例性QUERY块适合于答复John Doe批准了什么销售这个问题。

[0043] select*

[0044] from orders

[0045] where approver_name='John Doe'

[0046] 可以通过分析基于QUERY块的一个或多个查询的结果来答复时间性查询，如果该一个或多个查询是在开始事件和结束事件之间发出的话。这里，该一个或多个查询被称为基本查询。虽然BEGIN QUERY END块具有与SQL类似的格式，但这并不是必要条件。

[0047] 确定数据库事件何时发生的过程流程

[0048] 图1是示出用于检测数据库中何时发生事件的过程100的流程图。在一个实施例中，响应于接收到时间性查询而发起过程100。将利用下述示例来论述过程100：用户希望知道在John Doe被错误地给予购买订单批准权力的角色的时间期间，John Doe批准了什么销售。以上BEGIN和END块将用于帮助说明过程100。因此，过程100可用于确定何时发生了事件，例如John Doe被给予具有购买订单批准权力的角色。如前所述，BEGIN块描述了此条件。但是，过程100所检测的事件不限于时间性查询的开始事件。更一般性地，过程100可用于检测任何指定事件何时发生于数据库中。图2的过程200和图3的过程300提供了确定对时间性查询的答复的细节，例如John Doe批准了什么订单。

[0049] 在图1的步骤102中，将数据库的至少一部分恢复到所调查的事件之前的时间。在一些情况下，可以假定适当的时间。如果事件是John Doe被赋予购买订单权力的角色的时间，则适当的时间可以是John Doe的雇用日期。如果没有到达更新近的时间，则可以恢复数据库的最老版本。不需要恢复整个数据库。例如，在销售人员示例中，有关信息将是销售表或角色表。因此，这些表被恢复。对销售或角色表的访问可取决于其他对象(从属对象(dependent object))，例如索引。因此，从属对象也可能需要被恢复。这里使用的术语“从属对象”指的是任何用于帮助访问或操纵另一数据对象或者更高效地进行该操作的对象。关于限制数据库的被恢复的部分的更多细节在下文中论述。

[0050] 在步骤104中，创建一个或多个数据库触发以检测BEGIN块中的条件(开始事件)。在DBMS中，触发是指定在特定事件发生时将被自动执行的一系列动作的对象(例如过程)。由触发指定的该系列动作一般被用诸如SQL或PL/SQL(可从Redwood Shores，California的Oracle公司获得的SQL的过程语言扩展)之类的高级数据库语言写为指令。例如，在关系数据库中，触发可被设计为在表视图或者数据库表的行被更新、插入或删除时启动。因此，触发一般与单个数据库表相关联。利用销售人员示例，有关的表是BEGIN块中指定的角色表。因此，可以为角色表创建数据库触发。如果BEGIN块参考了多个表，则可以为每个表创建数据库触发。

[0051] 可以为其他表创建数据库触发。例如，用户可以在BEGIN块中指定挂钟时间，而不是指定事件。因此，在销售人员示例中，没有针对角色表设定的触发。但是，可以针对具有挂钟时间和DBMS所保持的逻辑时间之间的转换的表设定触发。例如，特定的表可以在真实或挂钟时间的每100秒左右记录SCN(系统改变号)或其他逻辑时间。该表不一定具有每个SQL语句被提交时的确切挂钟时间，但可以提供非常接近的近似。触发被设定来确定何时达到(或近似达到)挂钟时间。

[0052] 可以优化触发，以规定仅当BEGIN块中指定的列被修改时触发才被执行。在销售人员示例中，BEGIN块中的列是“name”(姓名)和“role_name”(角色名)。一般地，在步骤104中生成的触发可以是将会检测BEGIN块中的条件何时取值为TRUE的任何机制。

[0053] 实现触发，以在检测到BEGIN块中的条件时发生所需的动作。例如，所需的动作可以是确定对时间性查询的响应。数据库触发的实现取决于下文论述的许多因素。在一个实施例中，数据库触发是通过检查END块来实现的。如果已知BEGIN块等于END块，那么可以基于QUERY块本身来实现触发的主体，以便答复时间性查询。但是，如果存在下述时间范围(END-BEGIN)，则触发的主体可设定表明检测到开始事件的标志：在该时间范围期间，需要计算发出一个或多个基于QUERY块的查询的结果。使用标志的一个示例在图3的过程300中论述，其中标志被用于触发对物化视图的填充。

[0054] 在可选的步骤106中，创建触发以检测END块中的条件(“结束事件”)。如果过程100的结束目标是确定开始事件发生的挂钟时间，则不需要执行步骤106。但是，由于过程100可用作答复时间性查询的起始点，因此对结束触发的创建允许了对时间性查询的有关范围的确定。结束触发是根据所需的应用实现的，并且在下文中论述。

[0055] 在步骤108中，从数据库的恢复日志重建表示数据库操作的语句。在一个实施例中，从恢复日志重建SQL语句。例如，由Redwood Shores，Calif.的Oracle公司提供的Oracle LogMiner可用于从恢复日志生成SQL。但是，并不一定要求使用LogMiner。

[0056] 重建的SQL语句不一定是用于作出日志中记录的数据库改变的那个SQL语句。例如，存在许多不同的可能导致对数据库的特定改变的SQL语句，而恢复日志信息可能不能确定导致该改变的确切SQL语句。但是，事件仍可被检测。作为特定示例，许多不同的SQL语句可向John Doe赋予销售人员的角色。但是，过程100可检测对John Doe的销售人员角色的改变，而不考虑导致该改变的SQL语句。

[0057] 不需要针对与数据库相关联的所有恢复日志重建表示数据库操作的语句。例如，如果过程100被用于为示例性的时间性查询检测开始事件，则只有属于BEGIN块中的对象集合的恢复日志需要被处理。如果过程100被用于为示例性的时间性查询检测结束事件，则只

有属于END块中的对象集合的恢复日志需要被处理。如前所述，过程100不限于确定开始事件，而是更一般性地适用于事件检测。因此，更一般性地，针对其重建数据库操作的恢复日志是属于与被检测的事件相关联的对象的日志。例如，可以针对与以下各项相关联的对象来重建数据库操作：1)开始事件、2)结束事件、3)QUERY块、4)元数据，例如用于检测开始事件、结束事件和QUERY块的执行的目录对象和索引。

[0058] 在步骤110中，将数据库操作应用到重建的数据库，以向数据库应用改变。如前所述，只需要应用有关的改变。例如，只有与销售表、角色表或从属对象有关的改变被应用到数据库。

[0059] 如果在112中BEGIN触发启动，则在步骤114中，触发的主体中定义的触发动作被应用。如前所述，触发可执行多种动作。在一个实施例中，触发动作确定和报告事件何时发生。例如，通过将SCN(系统改变号码)与挂钟时间关联起来，来确定和报告John Doe被授予批准购买的权力的时间。在另一个实施例中，响应于开始触发启动，采取一系列动作来确定在开始事件之后直到另一时间为止对数据库作出的改变。因此，可以答复时间性查询。例如，DBMS确定在John Doe被授予购买订单权力之后直到该权力被删除为止Doe准予了什么购买订单。

[0060] 用于对时间性查询作出响应的过程流程

[0061] 在检测到开始事件之后，发出基于QUERY块的一个或多个查询并且分析结果。基本查询是针对数据库的一个或多个版本来执行的。基本查询的语义以及针对其执行基本查询的数据库的版本取决于：(a)QUERY块中的(一个或多个)语句，以及(b)在开始事件和结束事件之间数据库的行是否被更新或删除。

[0062] 用于对时间性查询作出响应的技术至少部分取决于基本查询是否是单调的(monotonic)。对于查询Qm，单调查询是这样一个查询，其中，对于t2>t1，Qm(t1)的结果集合是Qm(t2)的结果集合的子集或者等于Qm(t2)的结果集合。如果基本查询是单调的，并且不存在与时间性查询有关的行更新或删除，则可以针对结束事件时数据库的版本来执行单个基本查询。也就是说，恢复日志中的改变被应用到经恢复的数据库，直到结束事件被触发为止，并且执行QUERY块的结果可以通过在单个时间点-结束事件-执行QUERY块来确定。图2描述了对于此第一种情况的本发明的实施例。

[0063] 考虑确定John Doe批准了什么订单的示例。利用以下SQL语句可在结束事件处发出以下基本查询。

[0064] select*

[0065] from orders

[0066] where approvername=″John Doe″

[0067] 以上语句将会返回John Doe所批准的所有订单。对于该特殊情况，这提供了准确的结果。但是，如果基本查询不是单调的，或者存在行更新或删除，那么在结束事件处数据库的状态不包含用于答复时间性查询的所有信息。例如，如果在John Doe批准购买订单时一行被插入，但随后该行由于对订单的修改而被更新，那么更新后的行就不包含初始的订单信息。图3描述了本发明的使用物化视图来为这种更一般性的情况提供结果的实施例。

[0068] A)单调查询和无数据库删除/更新

[0069] 图2描述了根据本发明实施例的为时间性查询提供结果的过程200。关于销售人员

的示例性时间性查询被用于帮助说明过程200，但过程200不限于该示例。过程200可以响应于过程100中的BIGIN触发启动而被发起；但是，并不一定要求过程100用于发起过程200。另外，过程200的一些步骤可在触发启动之前执行。过程200可包括触发本身的一些或全部动作，但这并不是必须的。

[0070] 在步骤202中，确定与QUERY块相关联的基本查询是否是单调的。如果它不是单调的，则过程200退出，而不提供结果。如果需要，步骤202可以在触发启动之前执行。描述了用于将非单调查询转换成单调查询的技术。例如，Terry等人的论文“Continuous Queries over Append-Only Databases”描述了用于将非单调查询转换成单调查询的技术。如果形成来对用户的时间性查询作出响应的初始基本查询是非单调的，则可以将该基本查询转换成单调形式。但是，过程200可以不将基本查询转换成单调形式，而是退出。如果过程200退出，则图3中的过程300可被执行以提供结果。

[0071] 如果基本查询是单调的，那么在步骤204中，从针对数据库的恢复日志重建数据库操作。需要处理的恢复日志只有针对与QUERY块有关的对象、结束事件以及执行QUERY块或结束事件触发所需的诸如目录对象和索引之类的其他从属对象的恢复日志。例如，示例性的QUERY块指定订单表；因此，只有针对订单表、订单表上的任何索引、目录对象和角色表的恢复日志被处理。

[0072] 步骤206是确定是否存在对与基本查询有关的行的删除或更新。如果是，则过程200退出，而不为时间性查询提供结果。例如，如果存在对订单表的行的删除或更新，其中批准者姓名是John Doe，则过程退出。如果过程200没有结果地退出，则图3中的过程300可被执行以提供结果。

[0073] 在步骤208中，过程200继续，直到结束触发启动，以重建更多数据库操作并且确定是否发生有关删除或更新。因此，结束触发仅在基本查询为单调并且不存在有关更新或删除的情况下才启动。如果结束触发启动，则在步骤210中针对数据库执行基本查询。例如，可以执行以下基本查询以确定在John Doe被授予权力和该权力被取消的时间之间Doe批准了什么订单：

[0074] select*

[0075] from orders

[0076] where approver_name=″John Doe″

[0077] B)使用物化视图的一般性情况

[0078] 图3是示出根据本发明实施例的利用物化视图向时间性查询提供结果的过程300的步骤的流程图。销售人员的示例性时间性查询被用于帮助说明过程300，但过程300并不限于该示例。如前所述，如果图2的过程200退出而没有向时间性查询提供结果，则过程300可被执行。但是，过程300可以在没有首先执行过程200的情况下被执行。

[0079] 在步骤302中，创建视图和集结区域(staging area)。物化视图是这样一种视图，对于该视图，视图数据的拷贝与原先从中收集和得出数据的基本表是分开存储的。物化视图中包含的数据在这里被称为(“物化数据”)。对物化视图的使用消除了反复发出基本查询的必要。

[0080] 物化视图帮助识别由于开始事件和结束事件之间作出的改变而引起的基本查询的结果子集。物化视图可以在恢复日志被重新应用到数据库时被自动填充。但是，也可使用

其他解决方案。另外，不一定要求物化视图被自动填充。在一个实施例中，恢复日志是通过SQL层来应用的，从而物化视图被自动填充。

[0081] 物化视图是基于QUERY块创建的。根据一个实施例，物化视图具有布尔列。布尔列将存储下述时间窗口：在该时间窗口期间，视图中的结果行满足了时间性查询。以下是用于销售人员示例中的时间性查询的示例性物化视图。

[0082] create view orders_view

[0083] as select FLAG,o.*

[0084] from orders

[0085] where approver_name="John Doe"

[0086] 在示例性物化视图中，“FLAG”是全局状态变量，其在orders_view被创建时被设定到0。在达到BEGIN事件之后，FLAG被设定到1。可以通过对BEGIN触发的主体的执行来设定FLAG。

[0087] 继续说明步骤302，创建集结区域，用于存储在达到结束事件时将返回给用户的结果。在销售人员示例中，集结区域最初将是空的，因为假定销售人员在批准权力被授予的开始点尚未批准任何订单。很明显，取决于时间性查询的指定，可能存在开始点处集结区域不为空的情况。

[0088] 物化视图和集结区域被描述为分开的对象，以便于说明，以及再使用物化视图的实现。但是，可以为两者使用同一个对象。例如，可以为物化视图增加状态列，在其中它可充当集结区域。填充物化视图的代码应当被适当地修改以处理状态列。

[0089] 在步骤304中，从数据库的恢复日志重建表示数据库操作的语句。在一个实施例中，从恢复日志重建SQL语句。换言之，恢复是SQL层上的逻辑应用，而不是块级别上的物理应用。

[0090] 在步骤306中，将表示数据库操作的语句应用到数据库。在步骤308中，每当发生提交操作时，物化视图就被自动填充。

[0091] 如前所述，过程300适合于处理存在删除和更新操作并且/或者基本查询非单调的情形。步骤310被用于处理这些状况。在步骤310中，结果集结区域和物化视图中的行之间的集合差异被添加到集结区域。存在于物化视图中但不存在于集结区域中的任何行被拷贝到集结区域。利用销售人员示例，集结区域在销售人员被授予批准订单的权力的那个时刻是空的。因此，出现在物化视图中但未出现在集结区域中的行表明该销售人员在被授予这种权力之后批准了订单。

[0092] 任何存在于集结区域中但不存在于物化视图中的行都被保存在集结区域中，并且状态标志被设定到DELETED。利用销售人员示例，出现在集结区域但未出现在物化视图中的行可表明销售人员所批准的订单后来被处理并且该行后来因为订单处理完成而被删除。因为这与时间性查询有关，所以这种信息应当被返回给用户。实际上，对集结区域的修改在用户所关注的时期期间产生了一时态表(temporal table)。

[0093] 过程300继续评估更多恢复日志，直到在步骤312中达到结束触发为止。然后控制进行到步骤314，其中，位于集结区域中的结果被返回给用户。然后过程300结束。

[0094] 不使用物化视图的替换实施例

[0095] 并不一定要求在过程300的步骤302中创建的视图是物化视图。取代使用物化视

图，QUERY块中指定的基本查询可被直接执行。为了使基本查询只返回在开始事件之后修改的行，基本查询在开始事件被触发之后立即被执行并且结果被保存在单独的表中。然后，重做日志(redo log)被处理，并且在每次COMMIT(提交)之后，基本查询被直接执行，并且在前一步骤中保存的行的集合被从基本查询的结果集合中去除。该行集合构成了本来会被物化视图机制逐渐填充的行。在此方法中，通过在每次COMMIT操作之后重新执行QUERY块来模拟物化视图的功能。

[0096] 在执行基本查询之后，对照集结区域比较基本查询结果，以计算集合差异，该集合差异以与过程300的步骤310相类似的方式被添加到集结区域。

[0097] 限制数据库的什么部分被恢复

[0098] 如前所述，不要求数据库的所有部分都在向其应用恢复日志之前被恢复。一般地，BEGIN、QUERY和END块被分析，以确定需要被恢复的表的集合和从属对象。在销售人员示例中，只有订单和角色表需要被恢复。但是，任何可用于评估BEGIN、QUERY和END块的诸如索引之类的从属对象也应当被恢复。另外，编译和执行BEGIN、QUERY和END块所需的所有目录或字典对象也被恢复。在大多数数据库系统中，目录或字典对象属于特殊的架构，例如Oracle SYSTEM或SYS架构，并且可以很容易被识别。

[0099] 为了只恢复数据库的子集，应当满足一个或多个条件。第一个条件是数据库应当允许对表的子集的恢复。第二个条件是从恢复日志生成的表示数据库操作的语句(例如SQL)引用单个表内的列值。换言之，一般的SQL语句不应当引用多个表。但是，有可能原来导致改变的SQL语句确实引用多个表。

[0100] 为了说明生成引用单个表的SQL语句的恢复的类型，考虑在正常运行时期间发出的以下简单SQL语句：

[0101] insert into order(myseq.nextval,...)

[0102] 在以上SQL语句中，“myseq”是SEQUENCE变量。在恢复期间，找出重做日志并且生成以下SQL语句：

[0103] insert into orders(500,...)

[0104] 注意，在恢复期间生成的SQL语句不取决于“myseq”。因此，即使在正向过程期间发出的SQL语句取决于“myseq”，在恢复期间发出的SQL也不取决于“myseq”。

[0105] 又例如，考虑这样一种情况，即，为了更新订单表，在正向过程期间发出的SQL语句引用从表(side table)。在这种情况下，在恢复期间生成的SQL不涉及从表。例如，在正向过程期间发出以下引用从表“emp”的SQL语句。

[0106] insert into roles

[0107] (select username

[0108] from emp

[0109] where empid=100),"approver"

[0110] 在正向过程期间，SQL语句使得表“emp”被访问，以便更新角色表。具体地，可以从emp表访问值“John Doe”。但是，在恢复过程期间生成的SQL不会引用“emp”表。例如，在恢复期间可生成以下SQL，其将具有与正向过程期间使用的SQL相同的效果。

[0111] insert into roles

[0112] ("John Doe","Approver")

[0113] 时间性查询的特殊情况

[0114] 用户发出的时间性查询不会总是指定开始事件和结束事件。例如，用户可以向BEGIN块或END块赋予挂钟时间，而不是将它们定义为事件的发生(即，条件变为TRUE)。数据库触发的性质可以取决于用户指定开始和结束事件的方式，如先前在图1的步骤104中所述。

[0115] 另一种情况是用户可能根本不提供关于开始或结束事件的任何信息。如果用户不指定开始事件，则数据库可被恢复到时间的起点或某个实际开始时间。如果结束事件未被指定，则结束事件可以默认为当前时间或另一实际结束时间。

[0116] 另一种情况是用户可能只希望确定何时作出了某个改变。在这种情况下，开始事件等于结束事件，并且对时间性查询进行答复的结果是一个值(例如，SCN或日志序列号)，该值可利用其他数据库信息与挂钟时间关联起来。

[0117] 另一种情况是用户可能希望在某个改变被作出时执行时间点查询。例如，如果列被错误地更新，则时间性查询可请求该错误更新之前的最新近的值。或者，时间性查询可以请求在指定改变被作出时的某个其他数据库状态。在这种情况下，就在BEGIN块所描述的改变被应用到数据库之前，基于QUERY块的查询将被执行。在这种情况下，在遇到开始事件时执行的触发的主体可能是QUERY块本身。

[0118] 硬件概述

[0119] 图4是示出本发明的实施例可在其上实现的计算机系统400的框图。计算机系统400包括用于传输信息的总线402或其他通信机构以及与总线402相耦合用于处理信息的处理器404。计算机系统400还包括诸如随机访问存储器(RAM)或其他动态存储设备之类的主存储器406，其耦合到总线402，用于存储信息和处理器404要执行的指令。主存储器406还可用于存储在处理器404执行指令期间的临时变量或其他中间信息。计算机系统400还包括只读存储器(ROM)408或其他静态存储设备，其耦合到总线402，用于存储静态信息和处理器404的指令。提供了诸如磁盘或光盘之类的存储设备410，其耦合到总线402，用于存储信息和指令。

[0120] 计算机系统400可以经由总线402耦合到显示器412，例如阴极射线管(CRT)，用于向计算机用户显示信息。包括字母数字和其他键的输入设备414被耦合到总线402，用于向处理器404传输信息和命令选择。另一类用户输入设备是光标控制装置416，例如鼠标、跟踪球或光标方向键，用于向处理器404传输方向信息和命令选择，并用于控制显示器412上的光标移动。该输入设备一般具有两个轴(第一轴(例如x)和第二轴(例如y))上的两个自由度，其允许设备指定平面中的位置。

[0121] 本发明涉及使用计算机系统400来实现这里描述的技术。根据本发明的一个实施例，这些技术由计算机系统400响应于处理器404执行包含在主存储器406中的一条或多条指令的一个或多个序列而执行。这种指令可以被从另一计算机可读介质(如存储设备410)读取到主存储器406中。对包含在主存储器406中的指令序列的执行使得处理器404执行这里描述的过程步骤。在替换实施例中，可以使用硬线电路来替代软件指令或与软件指令相组合以实现本发明。因此，本发明的实施例并不限于硬件电路和软件的任何特定组合。

[0122] 这里所用的术语“机器可读介质”指参与提供使得机器以特定方式工作的数据的任何介质。在利用计算机系统400实现的实施例中，例如，在向处理器404提供指令以供执行

时，涉及了各种机器可读介质。这种介质可以采取许多形式，包括但不限于：非易失性介质、易失性介质和传输介质。非易失性介质例如包括光盘或磁盘，如存储设备410。易失性介质包括动态存储器，如主存储器406。传输介质包括同轴电缆、铜线和光纤，包括含总线402的线路。传输介质也可以采取声波或光波的形式，例如在无线电波和红外数据通信期间生成的声波或光波。所有这种介质都必须是有形的，以使得介质所承载的指令能够被将指令读取到机器中的物理机构所检测到。

[0123] 机器可读介质的常见形式例如包括软盘、柔性盘、硬盘、磁带或任何其他磁介质，CD-ROM、任何其他光介质，穿孔卡、纸带、任何其他具有孔图案的物理介质，RAM、PROM和EPROM、FLASH-EPROM、任何其他存储器芯片或卡盘，下文中描述的载波，或者计算机可以读取的任何其他介质。

[0124] 各种形式的机器可读介质可用于将一条或多条指令的一个或多个序列传送到处理器404以供执行。例如，指令可以首先承载在远程计算机的磁盘上。远程计算机可以将指令加载到其动态存储器中，并利用调制解调器经由电话线发送指令。计算机系统400本地的调制解调器可以接收电话线上的数据，并使用红外发送器来将数据转换为红外信号。红外检测器可以接收在红外信号中携带的数据，并且适当的电路可以将数据置于总线402上。总线402将数据传送到主存储器406，处理器404从主存储器406取得指令并执行指令。主存储器406接收的指令可以可选地在处理器404执行之前或之后存储在存储设备410上。

[0125] 计算机系统400还包括耦合到总线402的通信接口418。通信接口418提供到与本地网络422相连接的网络链路420的双向数据通信耦合。例如，通信接口418可以是综合业务数字网络(ISDN)卡或调制解调器，以提供与相应类型电话线的数字通信连接。又例如，通信接口418可以是局域网(LAN)卡，以提供与兼容LAN的数据通信连接。也可以实现无线链路。在任何这种实现方式中，通信接口418发送并接收电的、电磁的或光信号，这些信号携带了代表各种类型信息的数字数据流。

[0126] 网络链路420一般通过一个或多个网络提供到其他数据设备的数据通信。例如，网络链路420可以通过本地网络422提供与主机计算机424或由因特网服务供应商(ISP)426操作的数据设备的连接。ISP426进而通过全球分组数据通信网络(现在通常称为“因特网”)提供数据通信服务。本地网络422和因特网428都使用携带数字数据流的电的、电磁的或光信号。经过各种网络的信号和在网络链路420上并经过通信接口418的信号(这些信号携带去往和来自计算机系统400的数字数据)是传输信息的载波的示例性形式。

[0127] 计算机系统400可以通过(一个或多个)网络、网络链路420和通信接口418发送消息并接收数据，其中包括程序代码。在因特网示例中，服务器430可以通过因特网428、ISP426、本地网络422和通信接口418发送针对应用程序的请求代码。

[0128] 接收到的代码可以在接收时被处理器404执行，和/或被存储在存储设备410或其他非易失性存储介质中以供以后执行。以这种方式，计算机系统400可以获得载波形式的应用代码。

[0129] 在以上说明书中，已参考对于不同实现方式可能不同的许多具体细节描述了本发明的实施例。因此，关于本发明是什么以及申请人希望本发明是什么的唯一和排他指示是根据本申请授权的那套采取其授权时的特定形式的权利要求，包括任何后续的更正。这里针对这种权利要求中包含的术语明确阐述的任何限定都应当决定这种术语在权利要求中

使用时的含义。因此，在权利要求中没有明确记载的限定、要素、性质、特征、优点或属性都不应当以任何方式限制这种权利要求的范围。因此，说明书和附图被认为是说明性的而不是限制性的。

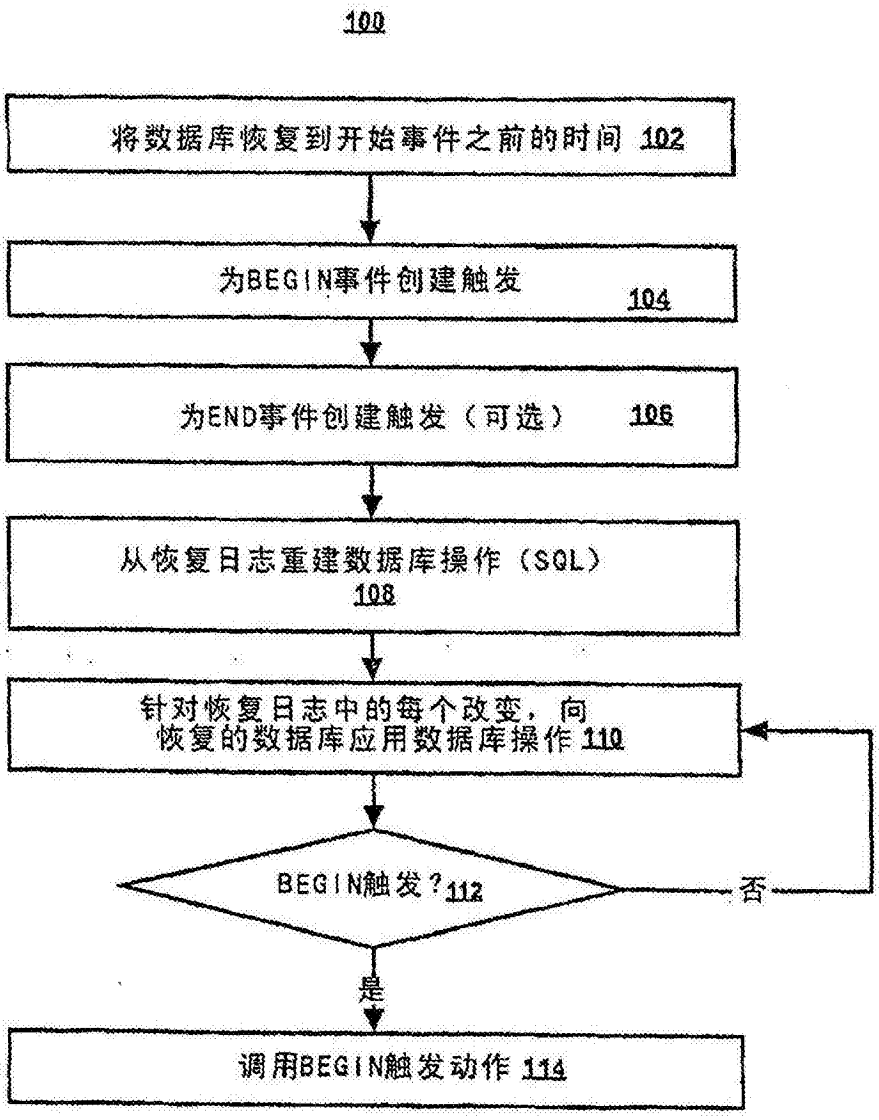

100
将数据库恢复到开始事件之前的时间 102
为BEGIN事件创建触发 104
为END事件创建触发（可选） 106
从恢复日志重建数据库操作（SQL） 108
针对恢复日志中的每个改变，向恢复的数据库应用数据库操作 110
BEGIN触发？112
否
是
调用BEGIN触发动作 114

图1

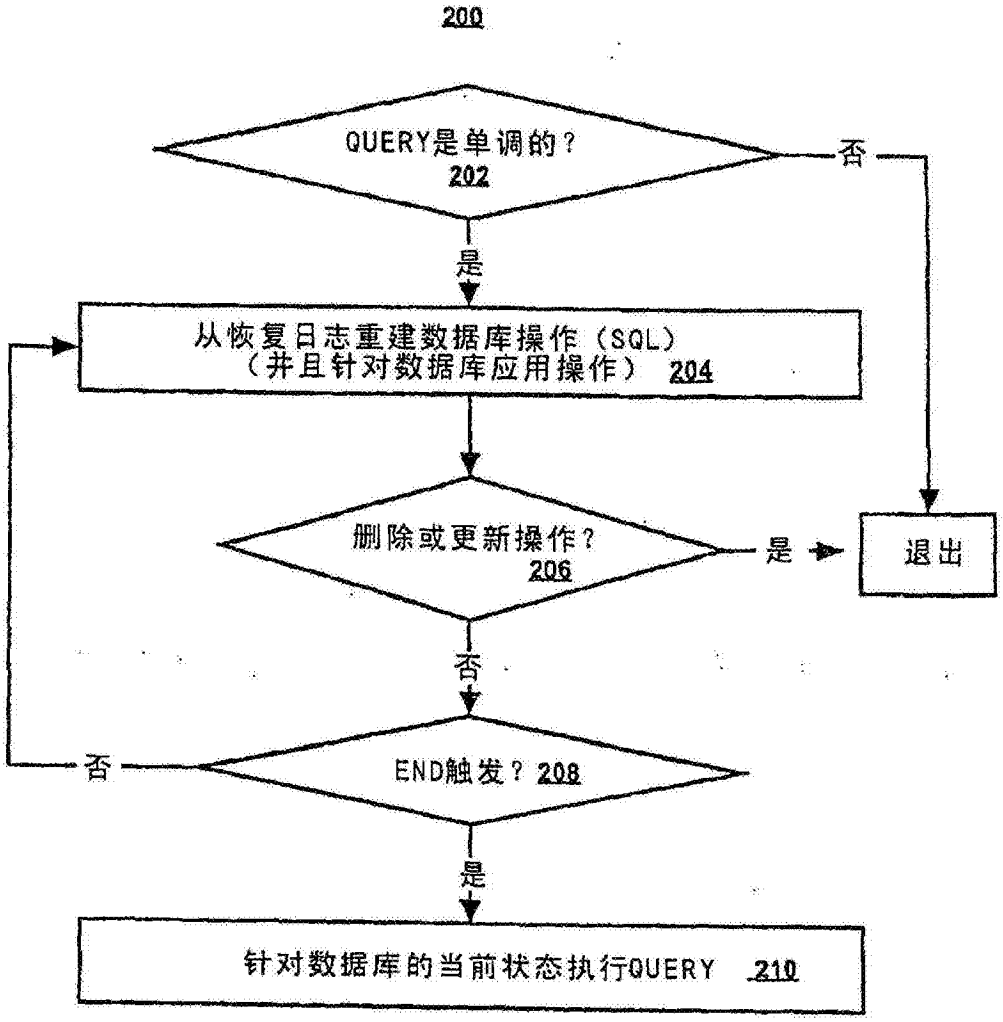

200
QUERY是单调的？
202
否
是
从恢复日志重建数据库操作（SQL）
（并且针对数据库应用操作） 204
删除或更新操作？
206
是
退出
否
否
END触发？ 208
是
针对数据库的当前状态执行QUERY 210

图2

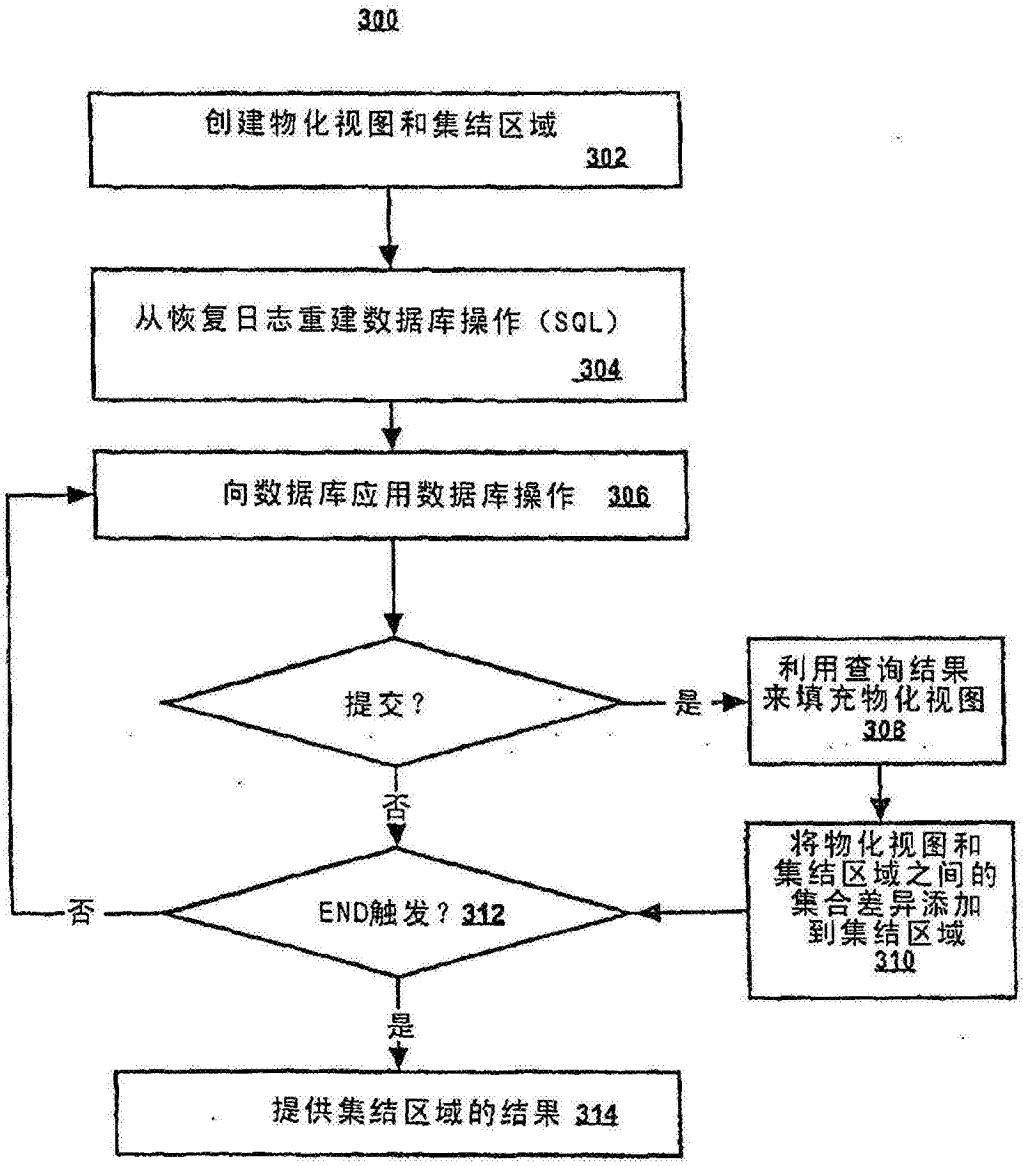

300
创建物化视图和集结区域
302
从恢复日志重建数据库操作（SQL）
304
向数据库应用数据库操作 306
提交？
是
利用查询结果
来填充物化视图
308
否
将物化视图和
集结区域之间的
集合差异添加
到集结区域
310
否
END触发？312
是
提供集结区域的结果 314

图3

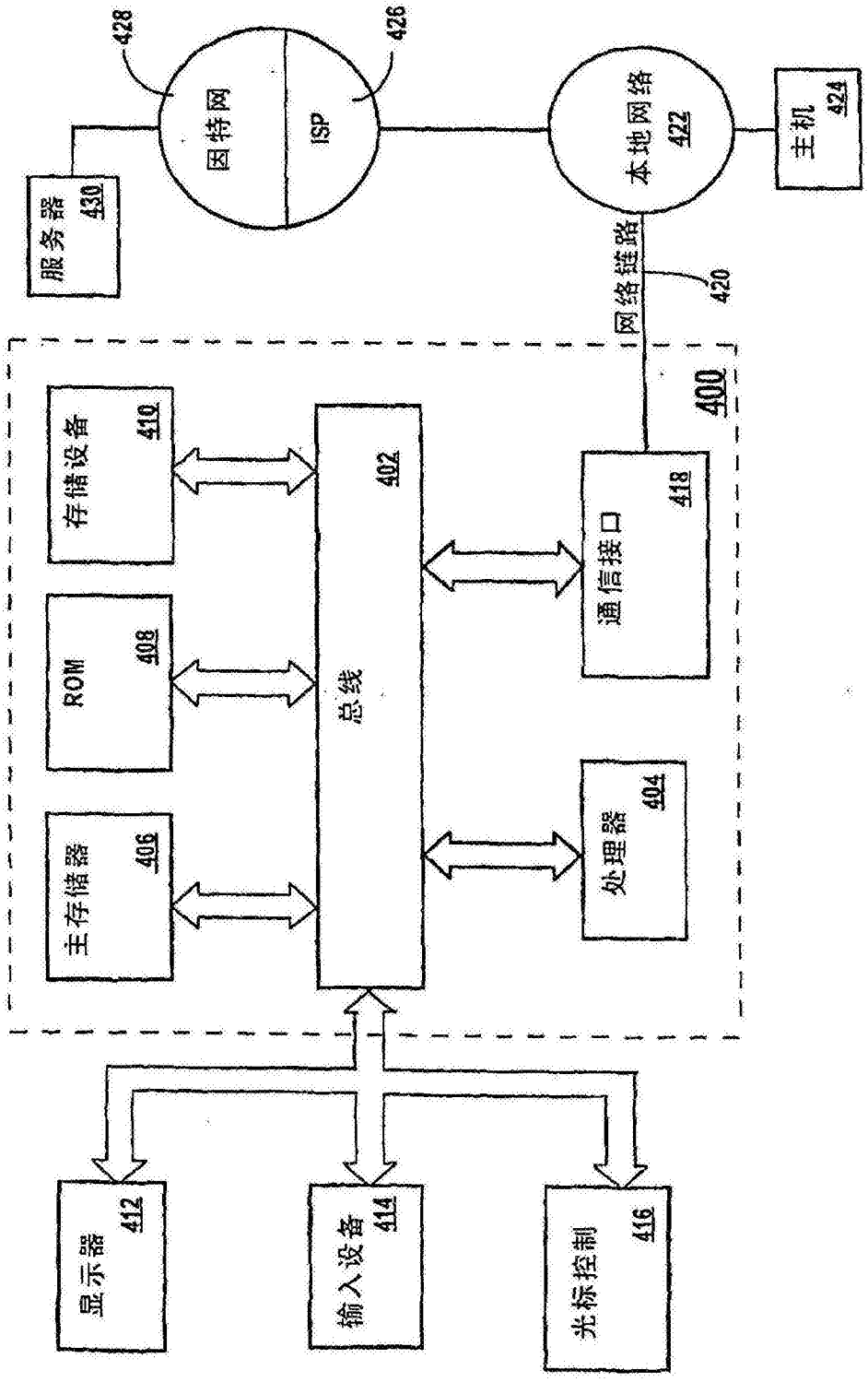

428
因特网
ISP
426
服务器
430
本地网络
422
主机
424
网络链路
420
400
存储设备
410
402
通信接口
418
ROM
408
总线
主存储器
406
处理器
404
显示器
412
输入设备
414
光标控制
416

图4