(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102279873 A
(43)申请公布日 2011.12.14

(21)申请号 201110168127.0
(22)申请日 2011.06.10
(30)优先权数据
12/813,577 2010.06.11 US
(71)申请人 微软公司
地址 美国华盛顿州
(72)发明人 W·E·艾特垦 N·阿维特尔
Q·布拉德利 B·洛夫林
S·J·米利特 B·奥兰尼科
P·F·礼萨伊 S·D·肯特
H·阿罕默德
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 钱静芳
(51)Int.Cl.
G06F 17/30(2006.01)

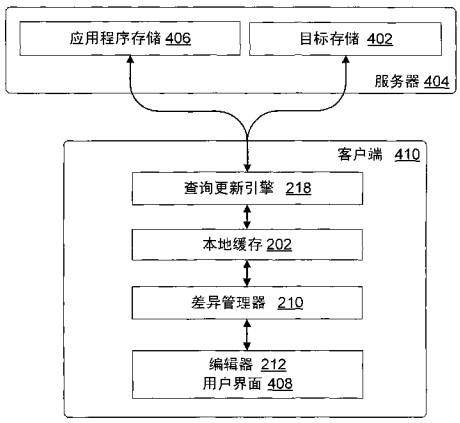
权利要求书 3 页 说明书 18 页 附图 4 页

(54)发明名称

对数据、模式，以及应用程序的统一并发变更

(57)摘要

在多用户数据库应用程序环境中管理变更。收集建议的变更，包括对数据、模式和/或应用程序描述的变更。用户可以指定属于这些类别中的一个或多个的项之间的外键关系。产生视图，该视图示出了建议的变更如果被成功地提交则将对该环境的影响。根据其依赖关系，对用户的建议的变更进行排序，并在单个事务中一起递交以供提交，服从乐观并发性和一致性检验。例如，对一个数据值的建议的变更可能与移除包含该数据值的数据元素的变更不一致。除指出提交是否成功之外，提交操作还可以返回标识符及其他返回值。



CN 102279873 A

1. 一种配置有数据以及指令的计算机可读非瞬态存储介质，所述指令当由至少一个处理器执行时使得所述处理器执行一种用于在多用户数据库应用程序环境中管理变更的过程，所述过程包括下列步骤：

收集 (302) 对所述多用户数据库应用程序环境的建议的变更，之后提交那些变更中的任一个；

对于特定的建议的变更 Y，标识 (304) 由 Y 假定的至少一个前提条件 PreY，以及通过提交 Y 所引起的至少一个后置条件 PostY；

对于特定的建议的变更 X，标识 (304) 由 X 假定的至少一个前提条件 PreX，以及通过提交 X 所引起的至少一个后置条件 PostX；

通过执行下列各项中的至少一项分析 (310) 变更 X 和变更 Y 的依赖关系：

判断 PostY 使 PreX 变得不可用，

判断 PostX 允许 PreY；

然后，

对所述建议的变更进行排序 (314)，以形成已排序的变更列表，在所述列表中按照提交次序，变更 X 在变更 Y 前面。

2. 如权利要求 1 所述的已配置的介质，其特征在于，所述排序步骤执行下列各项中的至少一项：

在所述已排序的变更列表中，把将创建数据库元素的建议的模式变更置于 (316) 将值写入该数据库元素中的建议的数据变更之前；

在所述已排序的变更列表中，把将引用数据库元素值的建议的变更置于 (316) 将移除该数据库元素的建议的变更之前。

3. 如权利要求 1 所述的已配置的介质，其特征在于，所述过程还包括在单个事务中提交 (320) 变更 X 和变更 Y 两者。

4. 如权利要求 1 所述的已配置的介质，其特征在于，所述过程还包括：

获取 (324) 对所述多用户数据库应用程序环境的一部分的独占访问，所述部分具有当获取所述独占访问时的当前状态；以及

对于每一个建议的变更，评估 (326) 该建议的变更的所有前提条件是否都存在于所述当前状态或者将由所述已排序的变更列表中的该建议的变更前面的建议的变更所引起。

5. 如权利要求 1 所述的已配置的介质，其特征在于，所述过程还包括：

基于至少两个已分析的依赖关系，按提交次序对至少三个建议的变更进行排序 (314)，所述建议的变更是下列变更类别中的至少两个：数据变更、模式变更、应用程序变更；

验证 (336) 每一个建议的变更的前提条件将在所述建议的变更的提交时间被满足；以及

在单个事务中，按提交次序提交 (320) 所述建议的变更，在所述单个事务中，将与所述建议的变更冲突的任何变更都被禁止。

6. 如权利要求 1 所述的已配置的介质，其特征在于，所述收集步骤收集，并且提交步骤提交下列变更类别中的每一个类别的至少一个建议的变更 (204)：数据 (130) 变更、模式 (128) 变更、应用程序 (122) 变更；并且其中，所述过程还包括从所述提交步骤获取 (322) 至少一个返回值。

7. 一种用于在多用户数据库应用程序环境中做出变更的过程，所述过程包括所述环境的特定用户的以下步骤：

观察 (338) 所述多用户数据库应用程序环境的值，并向编辑器递交对所述多用户数据库应用程序环境的至少某些观察值的多个建议的变更，所述建议的变更是下列变更类别中的至少两个：数据变更、模式变更、应用程序变更；以及

命令 (346) 所述编辑器将所有建议的变更一起提交，而不是命令所述编辑器提交所述变更中的一个，接收对该命令的响应，然后命令所述编辑器提交所述变更中的另一个，接收对该命令的响应，并继续逐个地提交所述变更。

8. 如权利要求 7 所述的过程，其特征在于，所述命令步骤命令所述编辑器提交下列变更类别中的每一个的至少一个建议的变更 (204)：数据 (130) 变更、模式 (128) 变更、应用程序 (122) 变更。

9. 如权利要求 7 所述的过程，其特征在于，所述过程还包括下列各项中的至少一项：

接收 (350) 对所述命令步骤的响应，所述响应指出变更数据值的建议的变更与 (a) 移除包含所述数据值的数据元素并且 (b) 在所述观察和递交步骤之后且在所述命令步骤之前作出的变更不一致；

接收 (350) 对所述命令步骤的响应，所述响应指出变更数据值的建议的变更与 (a) 移除包含所述数据值的数据库元素并且 (b) 在所述观察和递交步骤之后且在所述命令步骤之前作出的变更不一致；

接收 (350) 对所述命令步骤的响应，所述响音指出变更数据元素类型的建议的变更与 (a) 重命名所述数据元素并且 (b) 在所述观察和递交步骤之后且在所述命令步骤之前作出的变更不一致；

接收 (350) 对所述命令步骤的响应，所述响应指出变更数据库元素类型的建议的变更与 (a) 重命名所述数据库元素并且 (b) 在所述观察和递交步骤之后且在所述命令步骤之前作出的变更不一致；

接收 (350) 对所述命令步骤的响应，所述响应指出变更数据元素的应用程序显示规则的建议的变更与 (a) 移除数据库元素并且 (b) 在所述观察和递交步骤之后并且在所述命令步骤之前作出的变更不一致；

接收 (350) 对所述命令步骤的响应，所述响应指出变更数据元素的应用程序显示规则的建议的变更与 (a) 重命名数据库元素并且 (b) 在所述观察和递交步骤之后并且在所述命令步骤之前作出的变更不一致；

10. 如权利要求 7 所述的过程，其特征在于，还包括指定 (352) 属于下列类别中的一个或多个的项之间的至少一个关系：数据值、模式值、应用程序值；以及在所述递交步骤期间提供 (354) 所述关系。

11. 一种计算机系统，包括：

至少一个逻辑处理器 (110)；

可操作地与所述逻辑处理器进行通信的存储器 (112)，所述存储器位于至少一个机器中；

差异管理器 (210)，所述差异管理器驻留在所述存储器中，并且可操作以从用户姿势产生差异图 (208)；

用户专用变更缓存 (202)，所述用户专用变更缓存驻留在所述存储器中，并且可操作以为特定用户维护差异的集合，并为所述特定用户产生多用户数据库应用程序环境的视图 (206)，所述视图 (206) 反映零个或多个差异图如果被提交则对所述环境将具有的影响；以及

查询更新引擎 (218)，所述查询更新引擎驻留在所述存储器中，并且可操作以 (a) 访问下列类别中的每一个类别的值的共享多用户存储：数据值、模式值、应用程序值，(b) 将这些类别中的每一个类别的值变更递交到所述共享多用户存储，以及 (c) 检测将这些类别中的变更提交到所述共享多用户存储的尝试之后的结果。

12. 如权利要求 11 所述的系统，其特征在于，所述系统还包括驻留在所述存储器中的值变更 (204)，所述值变更包括由下列各项中的至少一个所提供的上下文中的 SQL 语句 (236)：乐观并发性检验、对与先前状态的一致性的检验。

13. 如权利要求 11 所述的系统，其特征在于，所述查询更新引擎可操作以基于建议的值变更之间的依赖关系 (216) 对建议的值变更进行排序，并且所述系统还包括值变更列表 (214)，所述值变更列表驻留在所述存储器中，并包含按适于在单个事务下提交的提交次序列出的每一个类别的建议的值变更。

14. 如权利要求 11 所述的系统，其特征在于，所述系统可操作以跟踪在多用户数据库应用程序的操作期间哪些值 (222，224，226) 被所述用户查询，并且所述查询更新引擎可操作以产生值变更，所述值变更是条件性的，在于只有在验证由所述用户查询的值没有被变更之后它们才请求提交。

15. 如权利要求 11 所述的系统，其特征在于，所述系统还包括：单个事务 (234)，所述单个事务包含对于数据库的值的值变更、对于所述数据库的数据库模式的值的值变更、以及用于访问所述数据库的应用程序的应用程序描述的值变更，这些值变更中的每一个都以同一个用户作为作者。

对数据、模式，以及应用程序的统一并发变更

技术领域

[0001] 本发明涉及多用户数据库，尤其涉及多用户数据库环境中的统一并发变更。

背景技术

[0002] 多用户数据库包括为多个人的一次或多次使用组织的数据。数据库可以按它们所包含的数据的种类，如书目、全文、数字、图像等等，来进行分类。数据库也可以根据它们用来组织数据并表示数据关系的数据库模型，诸如，例如，关系模型、分层模型、或网络模型，来进行分类。数据库模式以数据库管理系统 (DBMS) 支持的形式语言来描述数据库结构。例如，在关系型数据库中，模式可以定义表、字段、关系、视图、索引、封装、过程、函数、队列、触发器、类型、序列、物化视图、同义词、数据库链接、目录、及数据库的其他方面。模式有时被存储在数据字典中。

[0003] 数据库模式可以在对数据库的各种使用期间或在为这样的使用的准备中创建、访问和 / 或修改。例如，在数据库开发期间，可以使用模式来定义和组织该数据库被设计来容纳的数据的内容、关系和结构。在数据库询问期间，根据数据库的模式，用户访问数据库中的数据，以便进行信息检索和报表生成。在数据库维护期间，根据数据库的模式添加、删除或更新数据。数据库询问和数据库维护常常使用数据库应用程序来执行，该数据库应用程序可以是通用 DBMS 或专用的应用程序。在应用程序开发期间，模式支持例如数据输入屏幕、查询、表单、报表、表，以及标签的开发。

发明内容

[0004] 在多用户数据库应用程序的上下文中进行变更的用户可能面临许多不 确定性。例如，用户可能不确定已经变更了什么、数据库 / 模式 / 应用程序处于什么状态、不同种类的变更如何彼此进行交互、和 / 或由用户作出的变更如何与由其他用户作出的并发变更相关。在某些情况下，用户还要应付用于创建和保存不同类型的变更的不同用户体验模型。

[0005] 此处所描述的一些实施例帮助在多用户数据库应用程序环境中管理变更。一个实施例收集对多用户数据库应用程序环境的建议的变更，之后提交那些变更中的任一个。建议的变更可包括数据变更、模式变更、和 / 或应用程序变更。标识建议的变更的前提条件和后置条件，并使用这些条件来分析建议的变更之间的依赖关系。建议的变更根据其依赖关系按序放置在变更列表中。在验证每一个建议的变更的前提条件都将被满足之后，在单个事务期间按序提交建议的变更。如果建议的变更前提条件在当前状态中不存在，并且将不会由经排序的变更列表中的该建议的变更前面的建议的变更来提供，那么将引发错误并将其呈现给用户，且不提交建议的变更。除指示提交是否成功之外，提交还可以返回标识符及其他返回值。

[0006] 对于一些实施例，用户观察多用户数据库应用程序环境的值。用户向编辑器递交对至少一些观察值的多个建议的变更。建议的变更可涉及变更类别中的两个或三个，即，数据变更、模式变更、应用程序变更，并且用户还可以指定属于这些类别中的一个或多个的项

之间的外键和 / 或其他关系。用户命令编辑器将所有建议的变更一起提交，而不是试图逐个地提交变更，且伴随有中间结果以及其他用户可能作出的中间活动。用户接收对“提交所有变更”命令的响应，该响应指示任何建议的变更是否与在观察和递交步骤之后并且在命令步骤之前作出的另一个变更不一致。例如，对一个数据值的建议的变更可能与移除包含该数据值的数据元素的变更不一致，如果移除是在观察和递交步骤之后并且在命令步骤之前发生的话。也可以检测和报告其他不一致性。

[0007] 一些实施例包括可以从用户姿势产生差异图 (diffgram) 的差异 (diff) 管理器；差异图表示对观察数据、模式和 / 或应用程序值的建议的变更。用户专用变更缓存为特定用户维护差异图的集合，并产生多用户数据库应用 程序环境的视图，该视图反映了建议的变更差异图如果被提交则对环境将具有的影响。查询更新引擎可以访问数据值、模式值，以及应用程序值的共享多用户存储。这些类别中的每一个类别的值变更可被递交给共享多用户存储以便提交。值变更可以建议例如对数据库的关系值的变更和 / 或对应用程序描述的图对象的变更。值变更可包括乐观并发性检验或对与先前状态的一致性的检验的上下文中的 SQL 语句。提交尝试的结果可以由查询更新引擎来检测并报告回用户。

[0008] 所给出的示例只是说明性的。本发明内容并不旨在标识出所要求保护的主题的关键特征或必要特征，也不旨在用于限定所要求保护的主题的范围。相反地，提供本发明内容是为了以简化的形式介绍将在以下详细描述中进一步描述的一些概念。利用权利要求书定义本发明，在本发明内容与权利要求书有冲突的情况下，应该以权利要求书为准。

附图说明

[0009] 将参考附图给出更具体的描述。这些附图只示出了选定的方面，且因此不完全确定覆盖或范围。

[0010] 图 1 是示出一计算机系统并且还示出已配置的存储介质实施例的框图，该计算机系统具有至少一个处理器、至少一个存储器、多用户数据库应用程序环境、以及可以存在于多个网络节点上的操作环境中的其他项；

[0011] 图 2 是示出用于在图 1 操作环境中管理对数据、模式、以及应用程序值的统一并发变更的示例体系结构的各方面的框图；

[0012] 图 3 是示出某一进程的步骤和已配置的存储介质实施例的流程图；以及

[0013] 图 4 是示出一示例体系结构的框图，该示例体系结构包括差异管理器、查询更新引擎、以及客户端 - 服务器操作环境中的其他项。

具体实施方式

[0014] 概览

[0015] 多用户数据库应用程序的许多方面可以至少在理论上由应用程序的创建者或用户来变更。为清楚起见，这些方面可被归类为数据变更、模式变更、以及应用程序变更，但是，可以理解，某些变更跨越这些类别，并且在其他讨论中可以使用其他类别。应用程序变更涉及对数据库应用程序本身的描述（如页、布局等等）的变更。模式变更涉及对数据的描述（诸如，例如，表的模式）的变更。数据变更涉及应用程序的数据库的数据中的变更，如姓名、年龄、和 / 或存储在被应用程序访问的表中的其他数据。

[0016] 此处所描述的一些实施例提供了一种管理应用程序、模式、以及数据变更的统一的乐观并发性编辑模型。一些实施例向用户呈现统一用户模型中的编辑体验，并且还利用统一的体系结构来实现该解决方案。

[0017] 一些实施例在允许并发数据和模式变更的环境中使用乐观并发性。在某些实施例中，统一编辑体系结构支持某些实施例中的所有变更类别，而全局保存用户模型接收所有三个变更类别，并递交它们以便进行事务性提交。

[0018] 给定实施例还可以包括下列各方面中的一些。在某些实施例中，用户界面显示变更，并报告与其他用户的变更的交互。可以显示概述和状态指示。可以检测和解决冲突，包括例如模式－模式、模式－数据、应用程序－应用程序、应用程序－模式、以及应用程序－数据冲突。可以对图结构化数据定义乐观并发性方法。在建议的变更被实际应用之前，用户可以看到建议的变更的近似结果。在应用变更之前可以跨类别对变更进行排序，以减少错误并保留好的状态。

[0019] 现在将参考诸如附图中所示出的那些示例性实施例，并使用特定语言来对其进行描述。但是，精通相关技术的人员所能想到的对此处所示出的本发明的特点的更改和进一步的修改，如此处所示出的本发明的原理的其他的应用，都应该被视为在带有权利要求的本发明的范围内。

[0020] 在本发明中阐明了术语的含义，如此，应该在仔细关注这些阐明的情况下阅读权利要求书。给出了具体示例，但是，相关领域的技术人员将理解，其他示例也可以落在所使用的术语的含义范围内，并且在一个或多个 权利要求的范围内。术语不一定具有与它们在一般用途中、在特定行业的用途、或在特定词典或词典集中拥有的相同含义。附图标记可以与各种措词一起使用，以帮助显示术语的广度。从给定文本片段中省略附图标记不一定意味着没有通过文本讨论附图的内容。发明人声明并行使他们对他们自己的词典的权限。这里在详细描述中和／或在申请文件的别处显式地或隐式地定义了术语。

[0021] 如此处所使用的，“计算机系统”可包括，例如，一个或多个服务器、主板、处理节点、个人计算机（无论是否是便携式的）、个人数字助理、蜂窝或移动电话、和／或提供至少部分地通过指令来控制的一个或多个处理器的设备。指令可以采取存储器中的软件和／或专门电路的形式。具体而言，虽然许多实施例在工作站或膝上型计算机上运行，但是其他实施例也可以在其他计算设备上运行，并且任何一个或多个这样的设备都可以是给定实施例的一部分。

[0022] “多线程”计算机系统是支持多执行线程的计算机系统。术语“线程”应该被理解为包括能够或接受同步的任何代码，并且还可被称为另一名称，如“任务”、“进程”或“协同例程”。线程可以并行地、按顺序、或以并行执行（例如，多处理）和顺序执行（例如，时间分片）的组合运行。多线程环境是以各种配置设计的。执行线程可以并行地运行，或者线程可以被组织为并行执行，但是实际轮流按顺序执行。例如，多线程化可以通过在多处理环境中在不同的核上运行不同的线程、通过对单个处理器核上的不同线程进行时间分片、或者通过时间分片和多处理器线程化的某种组合来实现。线程上下文切换可以例如通过内核的线程调度器、通过用户空间信号、或通过用户空间和内核操作的组合来发起。线程可以轮流对共享数据进行操作，或者例如每一线程都可以对其自己的数据进行操作。

[0023] “逻辑处理器”或“处理器”是单个独立的硬件线程处理单元。例如，每个核运行两

个线程的超线程四核芯片具有八个逻辑处理器。处理器可以是通用的，或者针对特定用途，如图形处理、信号处理、浮点算术处理、加密、I/O 处理等等，对它们进行定制。

[0024] “多处理器”计算机系统是具有多个逻辑处理器的计算机系统。多处理器环境存在各种配置。在一给定配置中，所有处理器都在功能上是相等的，而在另一配置中，由于具有不同的硬件能力、不同的软件指派，或者两者，某些处理器可能不同于其他处理器。取决于配置，处理器可以在单条总线上彼此紧密耦合，或者它们可以是松散耦合的。在某些配置中，处理器共享中央存储器，在某些配置中，它们每一个都具有它们自己的本地存储器，而在某些配置中，存在共享的和本地存储器两种。

[0025] “内核”包括操作系统、系统管理程序、虚拟机，以及类似的硬件接口软件。

[0026] “代码”表示处理器指令、数据（包括常数、变量、以及数据结构），或者指令和数据两者。

[0027] “自动地”表示通过使用自动化（例如，通过软件为此处所讨论的具体操作而配置的通用计算硬件），而不是没有自动化。具体而言，“自动地”执行的步骤不是手动在纸上执行的或在人的心里执行的；它们是利用机器执行的。

[0028] 贯穿本文，可任选的复数的使用表示存在所指示的特征中的一个或多个。例如，“变更”表示“一个或多个变更”，或等效地“至少一个变更”。

[0029] 贯穿本文，除非明确地声明，否则任何对进程中的步骤的引用都假设该步骤可以由感兴趣的一方直接执行和 / 或由该方通过中间机制和 / 或中间实体间接地执行，且仍落入该步骤的范围内。即，由感兴趣的一方直接执行该步骤不是必需的，除非直接执行是明确地声明的必要条件。例如，涉及由感兴趣的一方执行的动作的步骤，如“传送到”、“发送到”、“递交到”、“提供到”或“传递到”目的地，可能涉及居间动作，如某个另一方执行的转发、复制、上传、下载、编码、解码、压缩、解压、加密、解密等等，但仍被理解为由该感兴趣的一方直接执行。

[0030] 每当提及数据或指令时，应该理解，这些项配置计算机可读存储器，从而将它变换为特定制品，而不是简单地例如存在于纸上、在人的脑子里、或作为线路上的瞬时信号。

[0031] 操作环境

[0032] 参考图 1，用于一个实施例的操作环境 100 可包括计算机系统 102。计算机系统 102 可以是多处理器计算机系统，或者也可以不是。操作环境在给定计算机系统中可包括一个或多个机器，它们可以是集群化的、以客户端 - 服务器方式联网的、和 / 或以对等方式联网的。

[0033] 人类用户 104 可以通过使用显示器、键盘、及其他外围设备 106 与计算机系统 102 进行交互。系统管理员、开发人员、工程师以及最终用户每一个都是特定类型的用户 104。代表一个或多个人操作的自动化代理也可以是用户 104。在某些实施例中，存储设备和 / 或联网设备可以被认为是外围设备。图 1 中未示出的其他计算机系统可以与计算机系统 102 进行交互，或者例如通过网络接口设备使用到网络 108 的一个或多个连接与另一系统实施例进行交互。

[0034] 计算机系统 102 包括至少一个逻辑处理器 110。计算机系统 102 与其他合适的系统一样，还包括一个或多个计算机可读非瞬态存储介质 112。介质 112 可以是不同的物理类型。介质 112 可以是易失性存储器、非易失性存储器、被安装就位的介质、可移动介质、磁

介质、光介质、和/或其他类型的非瞬态介质（而不是诸如只传播信号的线路之类的瞬态介质）。具体而言，诸如 CD、DVD、记忆棒、或其他可移动非易失性存储器介质之类的已配置介质 114 在被插入或以其他方式安装时可以在功能上变为计算机系统的一部分，从而使其内容可被访问以供处理器 110 使用。可移动的已配置介质 114 是计算机可读存储介质 112 的示例。计算机可读存储介质 112 的某些其他示例包括内置 RAM、ROM、硬盘、以及其他不能被用户 104 轻松地移走的存储设备。

[0035] 介质 114 被配置有可由处理器 110 执行的指令 116；“可执行”在此处按广义使用，以包括例如机器代码、可解释代码以及在虚拟机上运行的代码。介质 114 还被配置有数据 118，该数据通过指令 116 的执行被创建、修改、引用和/或以别的方式使用。指令 116 和数据 118 配置它们所在的介质 114；当该存储器是给定计算机系统的功能部件时，指令 116 和数据 118 还配置该计算机系统。在某些实施例中，数据 118 的一部分代表了诸如产品特征、库存、物理测量值、设置、图像、读数、目标、卷等等之类的现实的世界的项。这些数据也可以如此处所讨论的通过并发统一变更管理中的操作，例如，通过收集、分析、排序、提交、绑定、部署、执行、修改、显示、创建、加载和/或其他操作，来变换。

[0036] 多用户数据库应用程序环境 120，包括带有接口 124 的数据库应用程序 122、带有模式 128 和数据 130 的数据库 126、附图中所示出的其他软件及其他项，可以部分地或完全地驻留在一个或多个介质 112 内，从而配置那些介质。操作环境也可以包括诸如例如显示器 132、总线、电源、以及加速器之类的其他硬件。

[0037] 给定操作环境 100 可包括向开发人员提供一组协调的软件开发工具的集成开发环境 (IDE)134。具体而言，对于一些实施例的一些合适的操作环境包括或帮助创建被配置成支持程序开发的 Microsoft ® Visual Studio ®开发环境 (Microsoft Corporation 的标志）。一些合适的操作环境包括 Java ®环境 (Oracle America 股份有限公司的标志），并且一些操作环境包括使用诸如 C++ 或 C#（“C-Sharp”）之类的语言的环境，但是，此处的教示适用于各种各样的编程语言、编程模型和程序，以及使用多用户数据库应用程序环境或其组件的软件开发本身的领域以外的尝试。

[0038] 在图 1 中以轮廓形式示出了各项以强调它们不必是所示出的操作环境的一部分，但是可以与操作环境中的项进行交互操作，如此处所讨论的。未采用轮廓形式的项在任何附图或任何实施例中也不一定是必需的。

[0039] 系统

[0040] 图 2 示出了适合与一些实施例一起使用的体系结构的各方面。变更缓存 202（有时在客户端 - 服务器实施例中被称为本地缓存）包含对应用程序 122、模式 128、和/或数据 130 的建议的变更 204。变更缓存 202 也可以包含描绘应用建议的变更 204 的近似结果的视图 206。这样的视图 206 是近似的，在于它基于以前的时间点的环境 120 的状态，并且自从该时间点以来，其他用户可能不仅已经建议了变更，而且还提交了它们。

[0041] 在某些实施例中，建议的变更 204 被表示差异图 208，即，表示跨变更类别的一组编辑的数据结构。差异图可以使用树、列表、位标志、属性、对象、类、类方法、和/或适用于表示对应用程序 122、模式 128 和/或数据 130 的编辑的其他熟悉的数据结构来实现。在某些实施例中，差异图是从用户 104 向差异管理器 210 的用户界面输入的用户姿势中产生的。

[0042] 更一般而言，编辑器 212 接受作为用户输入的建议的变更 204，并将建议的变更给予数据库管理系统和 / 或应用程序编辑系统以供提交。在某些实施例中，编辑器将建议的变更 204 放在列表 214 中。在某些情况下，列表 214 是根据建议的变更之间的依赖关系 216 来排序的，以使得放置变更供提交的次序不一定是用户 104 在编辑器 212 中输入变更的次序。

[0043] 在某些实施例中，编辑器 212 包括差异管理器 210、变更缓存 202、以及查询更新引擎 218。在某些情况下，编辑器 212 的这三个组件使用差异图 208 彼此进行通信，如此，该体系结构相对于变更生命周期内的变更的内部表示是统一的。在某些实施例中，差异图可以表示所有三个类别（应用程序、模式、数据）的变更 204，如此，该体系结构相对于涉及应用程序 122、模式 128、以及数据 130 中的两个或更多的变更的内部表示是统一的。

[0044] 在某些实施例中，查询更新引擎 218 可以访问表示应用程序 122 的应用程序值 222、表示模式 128 的模式值 224、以及表示数据 130 的数据值 226 的共享多用户存储 220。就这一点而言，该体系结构相对于被单个编辑器 212 访问的单个共享存储中的所有三个类别的表示是统一的，该单个编辑器 212 接受三个类别（应用程序、模式、数据）中的每一个中的建议的变更 204。在某些实施例中，对数据库 126 的变更 204 可包括对数据库关系值 228（即，关系型数据库 126 模式 128 和 / 或数据 130 值）的变更。在某些实施例中，对应用程序 122 的变更 204 可包括对应用程序描述 232 中的图对象 230 的变更。

[0045] 在某些实施例中，将收集的建议的变更 204 放在单个值变更事务 234 中以便一起提交。这样的事务可包括一个、两个或所有三个变更类别（应用程序、模式、数据）的建议的变更 204，取决于用户的愿望以及实施例对 于统一变更的支持。一些实施例将至少某些变更作为 SQL 语句 236 实现以供提交。在某些情况下，在某些实施例中，SQL 语句可以在一个或多个乐观并发性检验 238 和 / 或对与先前的观察值的一致性检验 240 的上下文中发生。提交操作提供返回 242，在某些实施例中，返回 242 可包括由数据库管理系统所生成的标识符，以及成功 / 失败 / 错误代码返回。然后，编辑器 212 通过错误报告、成功消息等等来向用户 104 提供对应的响应 244。在某些实施例中，提交涉及丢弃成功提交的差异图，并且还重新同步高速缓存的值以获取提交后的高速缓存的值的当前状态。

[0046] 参考图 1 到 3，一些实施例提供了具有逻辑处理器 110 和存储器介质 112 的计算机系统 102，该逻辑处理器和存储器介质通过电路、固件和 / 或软件配置并位于至少一个机器中以便通过如此处所描述的对数据、模式、以及应用程序的统一并发变更来变换用户姿势和现有环境 120。在某些实施例中，驻留在存储器中的差异管理器 210 可操作以从用户姿势产生差异图 208。驻留在存储器中的用户专用变更缓存 202 可操作以为特定用户 104 维护差异图 208 的集合，并为特定用户 104 产生多用户数据库应用程序环境 120 的视图 206，视图 206 反映零个或多个差异图 208 如果被提交则对环境将具有的影响。驻留在存储器中的查询更新引擎 218 可操作以 (a) 访问下列类别中的每一个类别的值的共享多用户存储 220 ：数据值、模式值、应用程序值，(b) 将这些类别中的每一个类别的值变更 204 递交到共享多用户存储，以及 (c) 检测将这些类别中的变更提交到共享多用户存储的尝试之后的结果。

[0047] 在某些实施例中，一体系结构支持其中所有这些组件 202、210、218 在单个客户端可执行程序内实现的实现方式。一些实施例还支持这样的基于 web 的实现方式 ：其中，查询更新引擎 218 在中间层实现，而其他两个组件 210，202 在 web 浏览器中实现。

[0048] 一些实施例包括驻留在存储器中的值变更 204，该值变更包括由检验所提供的上下文中的 SQL 语句 236。例如，该上下文可包括乐观并发性检验 238，以查看特定用户的建议的值变更中涉及的值自从它们被当前用户最 后一次观察以来是否被另一个用户更改。SQL 语句上下文也可以或可另选地包括一致性检验 240，以检验建议的值与该值的先前状态的一致性。乐观并发性检验一般包括一致性检验，但是也可以检验其他值。例如，当变更数据值时，一致性检验可以将新的（建议的）数据值与最后一个观察值进行比较。并发性检查可以那么做，并且还检验模式中的可能影响数据值的变更。

[0049] 在某些实施例中，SQL UPDATE 语句 236 不受乐观并发性检验 238 的保护，但是接受一致性检验 240，因为 UPDATE 所应用到的集合受到声明将要被更新的行的先前状态的前提条件的限制。在更新之后，该实施例对已经受影响的行进行计数，并将此计数与本应受影响的期望行数进行比较。如果该数量不相等，那么，该实施例引发并发性错误并回滚整个事务。

[0050] 在某些实施例中，查询更新引擎 218 可操作以基于建议的值变更之间的依赖关系来对建议的值变更进行排序。一些实施例包括驻留在存储器中的值变更列表 214。列表 214 包含一个或多个（在某些情况下，所有三个）变更类别（数据、模式、应用程序）的建议的值变更；变更按适于在单个事务下提交的提交次序列出。在某些情况下，例如，单个事务更新一台机器上的单个存储，而在其他情况下，事务可以分布在多个存储和 / 或多个机器上。

[0051] 在某些实施例中，该系统的应用程序处理器或其他组件可操作以跟踪在多用户数据库应用程序 122 的操作期间哪些值被用户 104 查询。查询更新引擎 218 可操作以产生值变更 204，值变更 204 是条件性的，这表现在只有在验证由用户查询的值没有被变更之后，即，在一致性检验 240 之后，它们才请求提交。

[0052] 在某些实施例中，查询更新引擎可操作以产生对于数据库 126 的关系值 228 的值变更 204，且这样的变更驻留在系统中。在某些实施例中，查询更新引擎可操作以产生对于应用程序描述 232 的图对象 230 的值变更 204。在这些示例中，关系值 228、数据库 126、图对象 230、以及应用程序描述 232 可以各自通过熟悉的技术来提供。

[0053] 在某些实施例中，该系统包括单个事务 234，该单个事务 234 包含对于数据库 126 的值的值变更 204、对于数据库的数据库模式 128 的值的值变更 204、以及对于用于访问数据库的应用程序 122 的应用程序描述 232 的值变更 204。这些值变更中的每一个都可以具有同一个用户 104 作为作者。

[0054] 在某些实施例中，诸如人类用户 I/O 设备之类的外围设备 106（屏幕、键盘、鼠标、图形输入板、话筒、扬声器、运动传感器等等）将可操作地与一个或多个处理器 110 和存储器进行通信。然而，一实施例也可以深嵌入在系统中，以便没有人类用户 104 直接与该实施例进行交互。软件进程可以是用户 104。

[0055] 在某些实施例中，该系统包括通过网络连接的多个计算机。网络接口设备可以使用例如诸如分组交换网接口卡、无线收发器或电话网络接口之类的组件提供对网络 108 的接入，并将存在于计算机系统中。然而，一实施例也可以通过直接存储器存取、可移动非易失性介质、或其他信息存储检索和 / 或传输方法进行通信，或者，计算机系统中的一实施例可以不与其他计算机系统进行通信即可操作。

[0056] 进程

[0057] 图 3 以流程图 300 示出了某些进程实施例。在某些实施例中，附图所示出的进程可以自动地执行，例如，通过几乎不要求同时的实况的用户输入的测试脚本以运用编辑器 212，该编辑器 212 进而递交多个变更类别的值变更 204。除非另外指明，否则进程也可以部分自动地而部分手动地执行。在一给定实施例中，可以重复进程的零个或多个所示出的步骤，有可能利用不同的参数或数据来操作。一实施例中的步骤也可以按照与图 3 中展示的自顶向下次序不同的次序来执行。步骤可以串行地、以部分重叠的方式、或完全并行地执行。遍历流程图 300 以指出在进程中执行的步骤的次序可以在进程的一次执行与该进程的另一次执行之间不同。流程图遍历次序也可以在一个进程实施例与另一进程实施例之间不同。各步骤还可以被省略、组合、重命名、重组、或以其他方式偏离所示出的流程，只要所执行的进程是可操作的，并符合至少一个权利要求。

[0058] 此处提供了帮助示出该技术的各方面的示例，但是在本文内给出的示例并未描述所有可能的实施例。实施例不仅限于此处所提供的具体实现、排列、显示、特征、方法或情形。给定实施例可包括例如附加的或不同的特征、机制、和 / 或数据结构，并可以以别的方式偏离此处所提供的示例。

[0059] 在收集步骤 302 期间，一实施例收集建议的值变更 204。步骤 302 可以使用适用于接收例如寻求对数据、模式、和 / 或应用程序值的变更的姿势的用户界面或其他机制来完成。对建议的变更的收集 302 可以将对当前值的观察（例如，通过询问）与由用户向编辑器递交建议的变更交错。在某些实施例中，查询更新引擎充当或包含应用程序处理器或在转发查询之前截取查询并跟踪用户对值的观察的其他组件。

[0060] 在某些实施例中，用户的观察不必完全自相一致。例如，用户 A 可能观察按姓氏按字母顺序排序的雇员数据库中的前十个雇员的数据。基于该信息，A 可以更新一些数据。同时，用户 B 进行了很多编辑，不仅对第一组十个雇员，而且还对后面的（第二）组十个雇员进行了编辑，并提交这些编辑。然后，用户 A 向下翻页，以观察第二组十个雇员。一些实施例不刷新用户 A 对前十个雇员值的观察，而其他实施例却刷新。简而言之，A 对数据库的查看可能不是在每一方面都是完全一致的。然而，当 A 试图提交变更时，无法接受的不一致性最终将被发现。

[0061] 在标识步骤 304，一实施例标识收集的建议的值变更的前提条件 306 和 / 或后置条件 308。步骤 304 可以使用专用代码和 / 或体现对变更的逻辑分析的电路来完成。例如，在行中输入数据值或为数据值指定显示格式之前，如果行还不存在，则应该创建行。此处还提供其他示例，但是示例并不是全面的；它们只提供本领域技术人员可以详细叙述的指南。

[0062] 在依赖关系分析步骤 310，一实施例分析建议的变更 204 的依赖关系。分析可包括确定 312 后置条件和前提条件的影响。例如，分析 310 可以确定一个建议的变更的后置条件会干扰另一个建议的变更的前提条件，如当重命名行（例如，从“Bob”重命名为“Robert”）会干扰变更该行中的数据值（例如，Bob 的邮寄地址）或显示该行的内容的时候。作为另一个示 例，分析 310 可以确定一个建议的变更的后置条件促进另一个建议的变更的前提条件，如当添加表会促进向表添加数据值以及以报表来打印该表的时候。步骤 310 可以使用专用代码和 / 或体现对变更依赖关系的确定以及分析的电路来完成。

[0063] 在包括一个或多个变更放置步骤 316 的建议的变更排序步骤 314 期间，一实施例放置由用户 104 建议的变更，以形成 318 变更列表 214。变更列表不一定在每个时间点都按

照提交次序，但是在被递交以供在单个事务 234 中提交之前将按提交次序来放置。变更可以例如基于后置条件和前提条件来排序 314，以使得被置于列表 214 中靠前的变更的后置条件满足或至少不违反前提条件。链表、树、以及其他熟悉的数据结构可以用于实现变更列表 214 以及适用于如此处所教导的建议的变更和变更列表 214 的细节的步骤 314，316。

[0064] 一些实施例不对缓存的 202 更新差异图进行排序，而是稍后，例如就在提交之前在查询更新引擎中对它们进行排序。编辑器用户界面可以实施一些排序。例如，如果用户希望将 Order.CustomerName（定单．顾客名）设置为还不存在的顾客，那么不在下拉列表中向用户提供该 CustomerName。可以明显地通知用户，他们正在对缓存的预期的数据库进行操作。当对用于生成 T-SQL 语句以提交 320 变更的差异图进行排序 314 时，一些实施例不一定利用可能由用户提供的任何次序。

[0065] 在变更提交步骤 320 期间，一实施例试图提交（或等效地，指示 DBMS 和 / 或其他软件试图提交）对数据、模式和 / 或应用程序值的建议的变更 204 的已排序列表 214。在步骤 320 期间，可以使用例如到熟悉的 DBMS 的熟悉的接口及其他值维护技术。

[0066] 在返回获取步骤 322 期间，一实施例从提交步骤 320 获取返回 242。返回可包括例如成功代码和 / 或错误代码。在某些实施例中，返回 242 也可以包括由数据库管理系统或存储所生成的标识符，例如，当创建表或其他模式元素时，或当添加行时。在步骤 320 期间，可以使用例如到熟悉的 DBMS 的熟悉的接口及其他值维护技术。

[0067] 在访问获取步骤 324 期间，一实施例获取对多用户数据库应用程序环境 120 的一部分的独占访问，诸如通过使用熟悉的技术来获取锁。

[0068] 在变更评估步骤 326 期间，一实施例就对前提条件的满足来评估建议的变更列表 214 的变更。步骤 326 可涉及例如在一实施例获取 324 的独占访问的上下文中的分析步骤 310。

[0069] 在例如可以在变更评估步骤 326 期间发生的未满足的条件定位步骤 328 期间，一实施例定位其前提条件未被满足的建议的变更。该实施例可以引发 330 错误 332，向用户 104 报告存在未满足的条件，以及或许还包括其特征。一些实施例还防止 334 建议的变更被提交。

[0070] 在前提条件验证步骤 336 期间，一实施例或许在假设锁定的情况下验证变更列表 214 中的每一个建议的变更的前提条件将在变更被提交时被满足。步骤 336 例如在所有前提条件都被满足的情况下可包括评估步骤 326。

[0071] 在值观察步骤 338 期间，一实施例的用户观察涉及环境 120 的数据 130、模式 128、和 / 或应用程序 122 方面的值。例如，用户可以询问数据库 126，将模式 128 或其一部分加载到模式编辑器中，或将应用程序描述或其一部分加载到应用程序描述 232 编辑器中。

[0072] 在观察跟踪步骤 340 期间，一实施例跟踪值观察步骤 338 期间的用户活动。步骤 340 可以使用用户界面垫片 (shim)、日志记录和 / 或适用于跟踪诸如例如在 122、126、128、130、228、230 和 / 或 232 处表示的那些之类的特定项的其他熟悉的跟踪机制来完成。

[0073] 在用户控制的递交步骤 342 期间，一实施例的用户例如通过在编辑器 212 的用户界面作出用户姿势 344 向该实施例递交建议的变更 204。

[0074] 在提交命令步骤 346 期间，一实施例的用户命令该实施例在单个事务内尝试提交建议的变更 204 的集合，并返回该尝试的结果。例如，由用户使用的编辑器 212 可以具有

“全局保存”按钮，当被用户点击时，该按钮操作以发起提交命令步骤 346。

[0075] 在响应接收步骤 350，一实施例的用户接收对提交命令步骤 346 的响应 244。响应可以通过使用例如适用于在某些实施例中所使用的乐观并发性 的熟悉的用户界面机制、错误代码，以及成功指示符来提供。

[0076] 在关系指定步骤 352 期间，一实施例的用户指定在建议的变更 204 中所涉及的数据 130、模式 128，和 / 或应用程序 122 项之间的关系 356。例如，用户 104 可以指定在给定页面上显示的所有名称都是雇员姓名。可以使用例如适用于在编辑器 212 内对数据 130、模式 128 以及应用程序 122 项进行统一处理的熟悉的用户界面机制，但是也可以使用熟悉的软件工具代替编辑器 212 来指定关系。

[0077] 在关系提供步骤 354 期间，一实施例的用户向编辑器 212 提供在建议的变更 204 中所涉及的数据 130、模式 128、和 / 或应用程序 122 项之间的一个或多个关系 356。如果例如用户在编辑器 212 中直接指定关系，则步骤 354 可以通过指定步骤 352 来完成。另选地，如果用户在别处例如通过使用熟悉的软件指定 352 关系，那么可以通过向编辑器 212 导入在该指定 352 过程中所创建的文件或文件集合来提供 354 关系。

[0078] 在某些实施例中，在处理应用程序描述 232 的过程中推断一些或所有这样的关系。例如，如果应用程序描述指示一实施例对表 Customers 查询雇员 Bob 和 Fred 的列 Name（姓名）和 Age（年龄），该实施例发现用户已经观察了表 Customers、列 Name 和 Age 的应用程序描述和模式值，以及数据值 Bob 和 Fred 中的特定关系。

[0079] 在差异图产生步骤 358 期间，一实施例从用户姿势 344 产生差异图 208。可以使用例如适用于统一地接收姿势控制数据 130、模式 128、和 / 或应用程序 122 项的熟悉的用户界面软件，以及捕捉对这样的项的建议的变更的状态的差异图 208 数据结构。

[0080] 在差异图维护步骤 360 期间，一实施例维护对应于还没有被递交以供提交 348 的一组建议的变更 204 的差异图 208 的集合。步骤 360 可包括诸如对通过差异图表示的建议的变更进行排序 314，将这些建议的变更放置 316 在列表 214 中，分析 310 建议的变更依赖关系，评估 326 建议的变更，和 / 或跟踪 340 用户对数据 130、模式 128、和 / 或应用程序 122 项的观察。

[0081] 在视图产生步骤 362，一实施例为用户产生视图 206，该视图 206 反映 在由一个或多个差异图所表示的建议的变更被成功地提交 320 的情况下环境 120 的状态。可以使用诸如 HTML、Windows Presentation Foundation 等等之类的用于创建 / 配置可视显示 132 的熟悉的用户界面工具，它们适用于将视图基于未提交的变更 204 以及底层存储 220。

[0082] 在存储访问步骤 364，一实施例例如通过使用熟悉的工具来读和 / 写存储，例如在编辑器 212 和 / 或查询更新引擎 218 的指导下，访问多用户存储 220。

[0083] 在条件性值变更产生步骤 366 期间，一实施例产生条件性值变更 204，即，以乐观并发性检验 238 和 / 或先前值一致性检验 240 为条件的变更。步骤 366 可以例如基于在事务 234 期间要执行的检验 238、240 在条件性 SQL 控制流语句的上下文中使用 SQL 语句 236 来完成。对于检验 238、240 的条件可以基于例如分析 310 和 / 或评估 326 自动地生成，和 / 或作为由用户 104 显式地提供的条件来生成。

[0084] 在递交步骤 368 期间，一实施例例如响应于提交建议的变更的集合的命令 346 递交条件性值变更 204 以供提交 348。

[0085] 在返回检测步骤370期间，一实施例检测从提交尝试的返回242。返回可包括例如成功/出错指示符，并且可包括通过数据库和/或通过其他软件所提供的标识符。

[0086] 下面将参考各实施例更详细地讨论前面的步骤和它们的相互关系。

[0087] 一些实施例提供了用于在多用户数据库应用程序环境中管理变更的进程。下面首先主要从实现方式的观点来看来描述该进程，但是，可以理解，此处还描述并隐式地/显式地讲述了从用户的观点来看的对应的进程。

[0088] 在某些实施例中，一进程包括收集302对多用户数据库应用程序环境的建议的变更，之后提交那些变更中的任一个。该进程对于特定厄建议的变更Y，标识304由Y假定的至少一个前提条件PreY，以及通过提交Y所引起的至少一个后置条件PostY，并对于特定的建议的变更X，标识304由X假定的至少一个前提条件PreX，以及通过提交X所引起的至少一个后置条件PostX。为分析310变更X和变更Y的依赖关系，该进程判断312PostY 使PreX变得不可用（即，变更Y的结果违反变更X的前提条件，如此，Y无法在X之前被提交）和/或判断312PostX允许PreY（即，变更X的结果不排除变更Y，如此，X可以在Y之前被提交）。然后，该进程对建议的变更进行排序314，以形成已排序的变更列表214，在该列表中，按照提交次序，变更X在变更Y前面。

[0089] 前面的实施例涉及收集302建议的变更204并在试图提交320它们之前对它们进行排序314。这些实施例不一定包括或排除此处所讨论的其他方面，如乐观并发性、事务性提交、或对于数据、模式以及应用程序变更的统一的用户体验。

[0090] 作为一个具体示例，在某些实施例中，排序步骤314在已排序的变更列表中，把将创建数据库元素的建议的模式变更204置于316将把值写入到该数据库元素中的建议的数据变更204之前。例如，变更可能试图在Employees表中创建Age列，然后向Employees表添加一行，该行包括Age值。

[0091] 作为另一个具体示例，在某些实施例中，排序步骤314在已排序的变更列表中，把将引用数据库元素值的建议的变更204置于316将移除该数据库元素的建议的变更204之前。例如，变更可能在删除Bob的行之前试图将Tasks.AssignedTo（任务．分配给）从Bob变更为Fred。另一个示例是试图在删除Bob的行之前从Employees表中的Bob的行复制一些东西的变更。

[0092] 在某些实施例中，该进程在一个事务中提交320变更的集合。例如，该进程可能在单个事务234中提交320上面所指出的变更X和变更Y。这些实施例将事务性提交添加到上面指出的各方面（收集变更并对它们排序）。这些实施例允许但是不要求乐观并发性，以及对于对数据、模式、以及应用程序的变更的统一用户体验。

[0093] 在某些实施例中，该进程获取324对多用户数据库应用程序环境的一部分的独占访问。所述的部分具有在获取该独占访问时的一些当前状态。对于每一个建议的变更204，该进程评估326该建议的变更的所有前提条件 是否都存在于当前状态或者将由已排序的变更列表中的该建议的变更前面的建议的变更所引起。

[0094] 这样的实施例将乐观并发性的方面添加到收集和排序变更的方面。例如，响应于“保存”按钮，一些这样的实施例锁定可以变更的任何东西（数据、模式、应用程序描述），如此，获取324独占访问。然后，这些实施例检验以查看当用户收集要递交的建议的变更时什么已经变更，即，该进程相对于用户首先看到的值评估326建议的变更。可能涉及两种前提

条件，即，当客户首先观察时什么处于环境 120 中，以及作为已排序的变更的结果将产生什么。在某些实施例中，如果满足所有前提条件，则该进程提交 320 建议的变更。然而，在某些实施例中以及在一些情况下，用户 104 可以让一实施例在编辑建议的变更时在后台执行这些获取 324 和评估 326 步骤，而不实际提交 320 变更，直到用户已准备好并显式地命令 346 该实施例提交变更。

[0095] 在某些实施例中，评估步骤定位 328 具有以下前提条件的建议的变更：该前提条件在当前状态中不存在，并且将不会由已排序的变更列表中的该建议的变更前面的建议的变更所引起。在这种情况下，该进程引发 330 一个错误，并例如通过从事务中删除它来阻止 334 建议的变更被提交。此活动认识到，乐观并发性在某些情况下太乐观，在于可以由其他用户作出冲突的变更。

[0096] 在某些实施例中，收集步骤 302 收集下列变更类别中的至少两个类别的建议的变更 204：数据变更、模式变更、应用程序变更。在某些实施例中，收集所有三个类别的建议的变更 204。就这一点而言，一些实施例提供一种形式的跨变更类别的统一的用户体验。在某些实施例中，变更类别之间的区别是分明的，这表现在给定变更可能只影响单个类别。在其他实施例中，这样的区别是模糊的，这表现在变更可能涉及两个或者甚至三个类别。在各类别之间可能存在形式关系，但是，不是在每个实施例中都必需的。

[0097] 在某些实施例中，应用程序描述 232 与数据值和模式值存储在同一个数据库中，以便于所有三个类别的或包括应用程序变更的任何两个类别中 的变更的事务性提交。在其他实施例中，应用程序描述被分开存储，而事务性提交利用熟悉的分布式事务技术来完成，例如通过锁定总的存储 220 的应用程序描述部分、锁定总的存储 220 的模式数据部分、提交应用程序变更、提交模式和 / 或数据变更、解除锁定总的存储 220 的应用程序描述部分、以及解除锁定总的存储 220 的模式数据部分。

[0098] 在某些实施例中，该进程基于至少两个已分析的依赖关系按提交的次序对至少三个建议的变更进行排序 314，建议的变更是下列变更类别中的至少两个：数据变更、模式变更、应用程序变更。该进程验证 336 在建议的变更的提交时间每一个建议的变更的前提条件将被满足。然后，该进程在单个事务中按提交的次序提交 320 建议的变更，在该单个事务中，将与建议的变更有冲突的任何变更都被禁止。如此，一些实施例包括收集变更、在试图提交它们之前对它们进行排序、乐观并发性（例如，验证 336 前提条件）、事务性的提交 320（“单个事务”）、以及对于数据、模式、以及应用程序变更的统一的用户体验（“至少两个变更类别”）。

[0099] 如上所述，事务可以是单个事务，即使涉及多个数据库或其他存储部分中的多个锁。通过在涉及单独的数据库时使用多个锁，可以提供事务的特征。例如，进程可以行使对存储的独占写控制（不管有多少数据库构成存储 220），能够回滚变更，等等。

[0100] 在某些实施例中，该进程不会从提交步骤获得返回值。在其他实施例中，该进程从提交步骤获得 322 至少一个返回值 242。返回值可以是状态代码和 / 或标识符。术语“标识符”此处对于返回值 242 广泛地使用，并且可包括数据库生成的 ID 或从提交操作返回的修改过的值。例如，当插入新顾客时，该顾客的 ID 可以由 DBMS 自动生成。一旦事务成功地完成，就检索并返回新自动生成的 ID。然而，返回 242“标识符”不仅限于键或其他单一值；它们还可以是任何数据库自动生成的或修改的单元格。

[0101] 现在转向用户的观点，一些实施例提供用于在多用户数据库应用程序环境 120 中做出变更的进程，其包括该环境的特定用户观察 338 多用户数据库应用程序环境的值，并向编辑器递交 342 对多用户数据库应用程序环 境的至少一些观察值的多个建议的变更。建议的变更可以是一个或多个变更类别，即，数据变更、模式变更、应用程序变更。用户还命令 346 编辑器将所有建议的变更一起提交 320，而不是命令编辑器提交其中一个变更，接收对该命令的响应，然后命令编辑器提交另一个变更，接收对该命令的响应，并继续逐个地提交变更。

[0102] 可以理解，对建议的变更的递交不一定要最好被视为离散的用户动作，而是可以被视为一系列用户动作。用户例如对数据库中的一些数据进行一次或多次观察，收集针对这些观察值的建议的变更的集合，然后要求原子地提交建议的变更。观察和收集动作可以交错，并且除针对在所有建议的变更已经被收集并作出提交它们的尝试之后由其他用户进行变更进行保护之外，一些实施例还针对可能在该进程的此部分期间发生的任何外部变更进行保护。

[0103] 在某些实施例中，该进程涉及接收 350 对命令步骤的响应，其指出建议的变更 204 与在观察和递交步骤之后并且在命令步骤之前作出的另一个变更不一致。例如，SQL 事务 234 可能失败，产生一致性错误，因为在此用户递交建议的变更的时间和此用户命令 346 建议的变更被提交的时间之间另一个用户变更了某种东西（数据、模式和 / 或应用程序描述）。

[0104] 作为一具体示例，一些实施例接收 350 对命令步骤的响应 244，其指出对数据值的建议的变更与在观察和递交步骤之后且在命令步骤之前作出的、移除包含该数据值的数据元素的变更不一致。例如，或许客户 B 希望变更 Bob 的年龄，但是，客户 A 已经从 Employees 表中移除了 Bob 的行。

[0105] 作为另一个具体示例，一些实施例接收 350 对命令步骤的响应 244，其指出变更数据值的建议的变更与 (a) 移除包含该数据值的数据库元素，并且 (b) 在观察和递交步骤之后且在命令步骤之前作出的变更不一致。例如，客户 B 希望变更 Bob 的年龄，但是，客户 A 已经从 Employees 表中移除了 Age 列。

[0106] 作为另一个具体示例，一些实施例接收 350 对命令步骤的响应 244，其指出变更数据元素类型的建议的变更与 (a) 重命名该数据元素并且 (b) 在观察和递交步骤之后且在命令步骤之前作出的变更不一致。例如，客户 B 希望变更 Bob 的年龄类型，但是，客户 A 已经将 Bob 的行重命名为“Robert”。

[0107] 作为另一个具体示例，一些实施例接收 350 对命令步骤的响应 244，其指出变更数据库元素类型的建议的变更与 (a) 重命名该数据库元素并且 (b) 在观察和递交步骤之后且在命令步骤之前作出的变更不一致。例如，客户 B 希望变更 Age 列类型，但是，客户 A 已经重命名了 Age 列。

[0108] 作为另一个具体示例，一些实施例接收 350 对命令步骤的响应 244，其指出提变更数据元素的应用程序显示规则的建议的变更与 (a) 移除数据库元素并且 (b) 在观察和递交步骤之后且在命令步骤之前作出的变更不一致。例如，客户 B 希望变更 Age 列值（数据元素）在应用程序 122 中如何显示，但是，客户 A 已经从模式中移除了 Age 列（数据库元素）。

[0109] 作为另一个具体示例，一些实施例接收 350 对命令步骤的响应 244，其指出变更数

据元素的应用程序显示规则的建议的变更与 (a) 重命名数据库元素并且 (b) 在观察和递交步骤之后且在命令步骤之前作出的变更不一致。例如，客户 B 希望变更 Age 列值在应用程序中如何显示，但是，客户 A 已经重命名 Age 列。

[0110] 还可能有许多其他不一致情形。例如，客户 A 希望将列 Age 的类型从串变更为整型，而客户 B 希望将 Bob 的年龄从“32”（串）变更为“34”（还是串）。此处所提供的不一致情形的示例和示例检验并不是所有实施例中的所有可能性的全面。

[0111] 更一般而言，一些实施例从编辑器 212 接收 350 响应，其指出未提交的建议的变更 204 与由环境的另一个用户递交的已提交的变更不一致。在 SQL 事务例如以一致性错误失败之后可以向用户显示详细信息。一种实现方式可以具体地报告事务为什么失败，并可以指出就此可以联系哪些其他用户，因为也涉及了他们的变更。

[0112] 一些实施例包括指定 352 属于下列类别中的一个或多个的项之间的至少一个关系 :数据值、模式值、应用程序值，并且还包括在递交步骤期间提供 354 关系。例如，一些实施例允许用户 104 在存储器中的不同数据值、模式值或应用程序值之间创建关系。这样的关系可以基于临时存储器内结构，因为用于表示这样的关系的存在的键可以是由数据库在提交时间自动生成的。作为提交变更的一部分，一些实施例试图确保在这些实体之间创建更加永久的关系。例如，当插入新顾客 (c1) 和该特定顾客 (c1) 的新定单 (o1) 时，o1 在存储器中链接到 c1。然而，如果 c1 的主键是数据库生成的值，则在实际插入 c1 之后在 o1 和 c1 之间建立更加永久的关系，并检索 c1 的自动生成的主键值作为返回 242 值。一旦检索了该返回值，一实施例可以自动地将 o1 的 CustomerId（顾客 id）列更新到 c1 的主键的更加永久的值，以使得 o1 正确地链接到 c1。一些实施例使用从提交返回的结果（存储生成的值）来更新在变更缓存中高速缓存的值。其他实施例在成功提交之后丢弃变更缓存的所有高速缓存的值，并重新查询以利用例如足以支持应用程序的当前页面的值重新填充变更缓存。

[0113] 已配置的介质

[0114] 一些实施例包括已配置的计算机可读存储介质 112。介质 112 可包括盘（磁盘、光盘，或别的）、RAM、EEPROM 或其他 ROM、和 / 或其他可配置存储器，特别包括非瞬态计算机可读介质（而不是有线和其他传播信号介质）。已配置的存储介质可以特别地是诸如 CD、DVD 或闪存之类的可移动存储介质 114。可以是可移动的或不可移动的，并可以是易失性的或非易失性的通用存储器可以被配置成以从可移动介质 114 和 / 或诸如网络连接之类的另一源中读取的数据 118 和指令 116 的形式使用诸如差异图 208、已排序的变更列表 214、所有三个类别的建议的变更 204、和 / 或带有全局保存命令 346 的编辑器 212 之类的项的实施例，以形成已配置的介质。已配置的介质 112 能够使计算机系统执行用于通过诸如乐观并发性、已排序的变更、跨各变更类别的统一体系结构和用户体验之类的方面以及此处所公开的其他方面来变换数据的进程步骤。如此，图 1 到 3 帮助示出了已配置的存储介质实施例和进程实施例，以及系统和进程实施例。具体而言，图 3 中所示出的，或此处以其他方式教导的进程步骤中的任一个可以被用来帮助配置存储介质以形成已配置的介质实施例。

[0115] 补充示例

[0116] 下面提供了更多细节和设计考虑。如同此处的其他示例，在给定实施例中，所描述的特征可以单独地使用和 / 或组合地使用，或根本不使用。

[0117] 那些本领域的技术人员将理解，实现细节可以涉及诸如特定 API 和特定示例程序之类的特定代码，且因此不必出现在每个实施例中。本领域的技术人员还将理解，在讨论细节时所使用的程序标识符和某些其他术语是实现专用的，且如此不必涉及每个实施例。尽管如此，虽然它们不一定需要出现在这里，但是提供了这些细节，因为它们通过提供上下文可以帮助一些读者，和 / 或可以示出此处所讨论的技术的许多可能的实现中的一些。

[0118] 在某些实施例中，包括图 4 中所示出的一些，一体系结构包括三个主要子系统 ：查询更新引擎 218、本地缓存 202、以及差异管理器 210。这三个组件使用差异图 208 来进行通信。差异图是对数据项 118 的一组编辑的具体表示。在差异图内，要变更的数据具有标识。标识被一般地操纵。对于给定的一条数据 118，标识足以将该数据映射回到存储，如此标识的形式取决于数据 118 的源。具体而言，数据库 126 的目标数据可以通过关系存储在一个或多个服务器 404 上的目标存储 402 中，而应用程序描述 232 包括存储在充当应用程序存储 406 的对象储存库中的图结构化对象 230。在此示例中，存储 402、406 共同形成多用户存储 220。标识的形式在数据 118 的这些形式之间不同。

[0119] 在某些实施例中，查询更新引擎 218 负责访问 364 各种各样的存储的数据 118(应用程序描述 232、模式 128、以及目标数据 130)，将对此数据的变更递交 368 到存储，并检测 370 和解释这些变更的失败。查询更新引擎 218 的一个作用是以存储 220 理解的术语创建和解释标识 (也称做“标识符”)。当对存储做出查询时，一实施例可以为返回 242 的数据生成标识。当作出了更新时，一实施例可以将一组差异图 208 转换为要应用于存储 220 的操作序列，它们可以涉及对变更重新排序 314，以适应由存储施加的语义约束。

[0120] 在支持乐观并发性的各实施例中，当实施例和 / 或用户正在准备更新 时，存储 220 可能已经以与给定用户的更新相冲突的方式被变更。在某些实施例中，借助于底层存储，检测这些情况，并在可操作的响应 244 中报告它们是查询更新引擎 218 的职责。此职责可以依赖于对底层数据的语义的认识，并可以紧密耦合到存储 220。例如，关系数据和图对象数据具有不同的冲突检测规则。一些实施例也被设计成能处理变更将在数据存储更新之间冲突的可能性。为支持检测这样的冲突，一些差异图 208 包括从存储中获取的旧值，以及新建议的值。

[0121] 在某些实施例中，本地缓存 202 维护 360 差异图 208 的集合，并产生 362 用于向用户显示的数据 118 的视图，该视图表示如果当前累加的未提交的变更 204 被成功地提交 320 到存储器则将是什么状态。在某些实施例中，这是由用户看到和操纵的数据视图。除存储数据 118 的原始值以及当前建议的值之外，此组件也可以呈现当前数据库 126 和应用程序描述 232 值，以便于冲突的演示。

[0122] 在某些实施例中，当用户选择提交他们的变更 204 时，所有变更都被封装，并作为单个更新递交到查询更新引擎 218。368 可能成功地完成，或者也可能失败。在失败的情况下，一些实施例接收有关存储的当前状态的足够信息，以允许它们构建冲突的演示。在某些实施例中，在另一个事务被调用之前，解决冲突的用户的姿势被反映在用户界面中，以便给出系统状态已经变更的视觉反馈。

[0123] 在某些实施例中，差异管理器 210 负责将用户姿势 344 转换为差异图 208。当由查询更新引擎 218 评估查询时，它们被注解，以使得各个结果的标识可用。差异管理器 210 使用这些标识来为接收到的特定用户姿势创建差异图，例如通过编辑器 212 用户界面 408。

在某些实施例中，给定用户姿势可以导致许多差异图的创建。

[0124] 在某些实施例中，在驻留在客户端 410 上的单个可执行程序内实现多个组件 218、202、210、212。一些实施例包括这样的 web 实现方式：其中，查询更新引擎 218 在中间层实现，而组件 202、210、212 在浏览器中实现。

[0125] 至于用户模型，在某些实施例中，用户模型是这样的：使得所有类型 的变更—应用程序、模式、以及数据—都共享统一的包罗万象的体验。这样的一个示例是全局保存的体验；可以做出不同类型的变更，然后，当用户敲击保存按钮作为命令 346 时，所有变更都立即应用。此方法允许用户导航遍及环境 120，作出完全不同类型的变更，然后在用户空闲时决定保存。这有助于避免迫使用户对于不同类型的变更通过不同的体验来保存的方法所施加的情形，并且不会让用户在保存之前作出若干类型的变更。这里所讨论的用户模型就“保存”变更的单一概念而言，提供了大得多的灵活性和简洁性。

[0126] 在某些实施例中，还收集和记录所有类别的变更，以使得用户可以在统一的位置查看所有变更。那里，用户可以断定什么已经变更，添加了什么，删除了什么等等，以帮助跟踪还有待于保存的活动。当用户使用全局保存按钮保存时，向他们示出冲突的所有配置，并且他们可以推理冲突的所有配置，以使得他们可以作出有关各种数据源内冲突和跨数据源冲突的解决选项的决策。这比不允许用户直接看到他们的模式变更例如如何影响他们的数据的方法提供更大的灵活性。此外，它还可以允许用户选择他们的冲突解决选项，而不是让他们的变更完全被拒绝，或盖写其他用户的变更。

[0127] 更一般而言，一些实施例提供对变更的排序。编辑器 212 收集和缓存很多变更（对数据、模式、对应用程序描述的变更）。直到客户敲击“保存”之前，这些变更不被提交到数据库。更新引擎以使得变更可以被提交到数据库（在单一事务中）并成功的方式来对这些变更进行排序。

[0128] 一些实施例提供乐观并发性。给定用户可以在单次遇到打开 (Open)…编辑 (Edit)…保存 (Save) 时作出数据、模式、以及应用程序描述变更。用户基于首先查询数据库以观察数据、模式、以及应用程序描述的当前状态来创作这些变更。为确保他们的更新不会打断应用程序，一实施例可以确保数据库和 / 或其他涉及的存储 220 在用户首先观察值的时间和他们敲击“保存”以提交 320 他们的各种变更的时间之间没有被变更。在运行编辑器 212 的过程中，一些实施例跟踪引擎 218 查询了什么数据 / 模式 / 应用程序 描述。在提交时间，编辑器 212 开发反映他们的建议的数据 / 模式 / 应用程序描述变更 204 的 T-SQL 更新语句。T-SQL 语句是条件性的，验证 336 到变更被提交时，数据 / 模式 / 应用程序描述的观察值保持不变。

[0129] 作为涉及模式和数据变更的具体示例，通过编辑器，用户在“Employees”表中创建新列“Age”。然后，用户在 Employees 表中添加一个新行，并为在此新行中的“Age”单元格提供值。“Age”列在数据库中还不存在，因为这些变更被缓存在编辑器 212 中，以使得当用户敲击“保存”时它们可以被一同应用到数据库。用户敲击“保存”。为使这些变更的 SQL 事务成功，对它们进行排序。首先，在“Employees”表中创建“Age”列。其次，向“Employees”中添加一个新行（包括“Age”值）。

[0130] 作为涉及数据变更的具体示例，假设 Bob 刚刚被解雇，需要将他从“人力资源”数据库中删除。用户从 Employees 表中删除带有主键“Bob”的行。用户进入 Tasks（任务）表，

并将 Bob 的任务重新分配给他的代替者“Fred”。假设 Tasks 表具有外键约束，声明 Tasks 表中的 AssignedTo（分配给）列必须匹配 Employees 表中的一些雇员的 EmployeeName（雇员姓名）字段。对于 Tasks 表中的“修复一些隐错”任务，用户将“AssignedTo”字段从“Bob”更新为“Fred”。用户敲击“保存”。为使这些变更的 SQL 事务成功，对它们进行排序 314。首先，对于“修复一些隐错”任务，将 Tasks.AssignedTo 从 Bob 变更为 Fred。其次，删除 Employees 表中的“Bob”行。

[0131] 作为涉及乐观一致性和数据变更的具体示例，假设用户 A 希望从 Employees 表中删除 Bob，因为他被解雇，而用户 B 希望将 Employees 表中的 Bob 的年龄从 42 变更为 43。在编辑器 212 中，用户 B 将“Employees”表中的“Bob”行的“Age”字段从 42 变更为 43。用户 B 没有敲击“保存”，而是暂时转向其他事情。为作出从 42 到 43 的变更，编辑器查询数据库中的 Bob 的年龄。作为缓存变更的一部分（“保存”之前），编辑器注意到它最后观察到了 Bob 的年龄为 42（数据库＝“HR”、表＝“Employees”，行＝“Name:Bob”，字段＝“Age:42”）。当用户 B 专注于其他事情时，用户 A 从 Employees 表中删除“Bob”行，并将此变更保存到数据库。用户 B 将注意力回到编辑器，仔细考虑从 42 到 43 的变更，然后敲击“保存”。SQL 事务失败，带有一致性错误。对于 42-43 变更的 SQL 语句利用乐观并发性检验保护 UPDATE：

[0132] delete from Employees where Name ＝′ Bob′ and Age ＝ 43；

[0133] if@@ROWCOUNT ＜＞ 1 raiserror(′ CONCURRENCY ERROR′ ,1,16)；

[0134] 作为涉及乐观一致性和模式变更的具体示例，假设用户 A 希望将“Employees”表中的“Age”列的名称变更为“EmployeeAge”。用户 B 希望将“Age”列的类型从“nvarchar”变更为“int”。在编辑器中，用户 B 变更“Age”列的类型，但是没有立即敲击“保存”。相反，用户 B 在 HR 数据库中做了一些相关的工作。为作出这一类型变更，编辑器查询 SQL 模式表。作为缓存变更的一部分（“保存”之前），编辑器注意到它最后观察到了“Employees”表中的类型“nvarchar”的“Age”列。同时，用户 A 将“Age”列重命名为“EmployeeAge”，并将此变更保存到数据库。用户 B 判断到时间了，并敲击“Save”，意图与其他 HR 数据库工作一起保存“nvarchar”-＞“int”变更。SQL 事务失败，带有一致性错误。″ nvarchar″ -＞″ int″ 变更的 SQL 语句利用乐观并发性检验保护 UPDATE。

[0135] 作为涉及乐观一致性和应用程序变更的具体示例，假设用户 A 希望向应用程序 122 添加一个规则，说“当显示 Age 值时，以红色呈现那些＞ 55 的值”。用户 B 从 Employees 表中完全删除了 Age 列。通过编辑器 212，用户 A 编辑应用程序描述以包括该新规则，并刷新应用程序，看出该规则具有所需的效果。应用程序描述被存储在 HR 数据库中。用户还没有敲击“保存”。为运行 HR 应用程序，编辑器处理应用程序描述。在这样做时，它注意到这一事实：应用程序描述要求对 Employees 表进行 SQL 查询。作为处理应用程序描述中的新的与 Age 相关的规则的一部分，编辑器还注意这一事实：该规则涉及 Employees 表中的 Age 列。同时，用户 A 从 Employees 表中删除 Age 列，并将此变更保存到数据库。然后，用户 B 敲击“保存”，意图是保存修改的应用程序描述，并对添加的 Age 值着色。SQL 事务失败，带有一致性错误。将更新应用程序描述的 SQL 语句包括乐观并发性保护。

[0136] 结论

[0137] 虽然具体实施例在此处被明确示出并描述为进程、已配置的介质、或系统，但是可以理解，对一种类型的实施例的讨论也一般性地延伸到其他实施例类型。例如，结合图 3 对

进程的描述也帮助描述已配置的介质，并帮助描述类似于结合其他附图所讨论的那些的系统和产品的操作。对一个实施例的限制也不一定适用于另一个实施例。具体而言，进程不一定仅限于在讨论诸如已配置的存储器之类的系统或产品时呈现的数据结构和方案。

[0138] 不是图中所示出的每一项都需要存在于每个实施例中。相反，实施例可以包含图中未显式地示出的项。虽然一些可能性在此处通过具体示例在文本和附图中示出，但是各实施例可以偏离这些示例。例如，一示例的具体特征可以被省略、重命名、以不同的方式分组、重复、不同地以硬件和 / 或软件实例化，或是在两个或更多示例中出现的特征的混合。在某些实施例中，在一个位置处示出的功能也可以在不同的位置处提供。

[0139] 通过附图标记参考了附图。在附图或文本中与给定附图标记相关联的措词中的任何显而易见的不一致性应该被理解为仅仅时拓宽该标记所引用的内容的范围。

[0140] 如此处所使用的，诸如“一”和“该”之类的术语是包括所指出的项或步骤中的一个或多个。具体而言，在权利要求书中，对一个项的引用一般表示至少一个这样的项存在，并且对一个步骤的引用表示执行该步骤的至少一个实例。

[0141] 标题只是为了方便；有关给定主题的信息可以在其标题指出该主题的部分外面找到。

[0142] 所提出的所有权利要求都是本说明书的一部分。

[0143] 尽管在附图中示出了并在上文描述了示例性实施例，但是，对于本领域的技术人员来说显而易见的是，在不偏离权利要求书中所阐述的原理和概念的情况下，可以进行很多修改。尽管用结构特征和 / 或过程动作专用的语言描述了本主题，但可以理解，所附权利要求书中定义的主题不必限于 权利要求书上面所描述的具体特征或动作。不一定在给定定义或示例中标识的每一个手段或方面都在每个实施例中存在或使用。相反，所描述的具体特征和动作是作为供当实现权利要求书时考虑的示例来公开的。

[0144] 落入权利要求书的等效方案的含义和范围内的所有改变应在法律允许的最大可能的范围内被权利要求书的范围所涵盖。

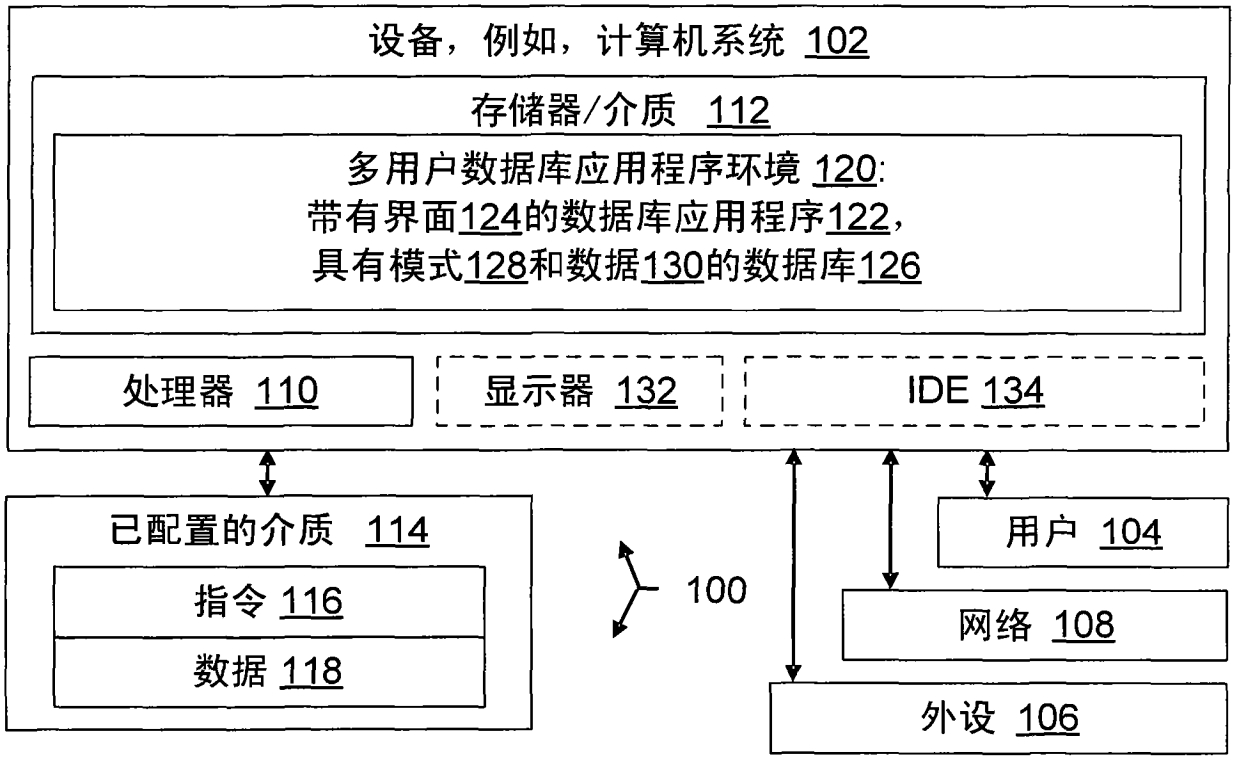

设备，例如，计算机系统 102
存储器/介质 112
多用户数据库应用程序环境 120:
带有界面124的数据库应用程序122，
具有模式128和数据130的数据库126
处理器 110
显示器 132
IDE 134
已配置的介质 114
指令 116
数据 118
100
用户 104
网络 108
外设 106

图 1

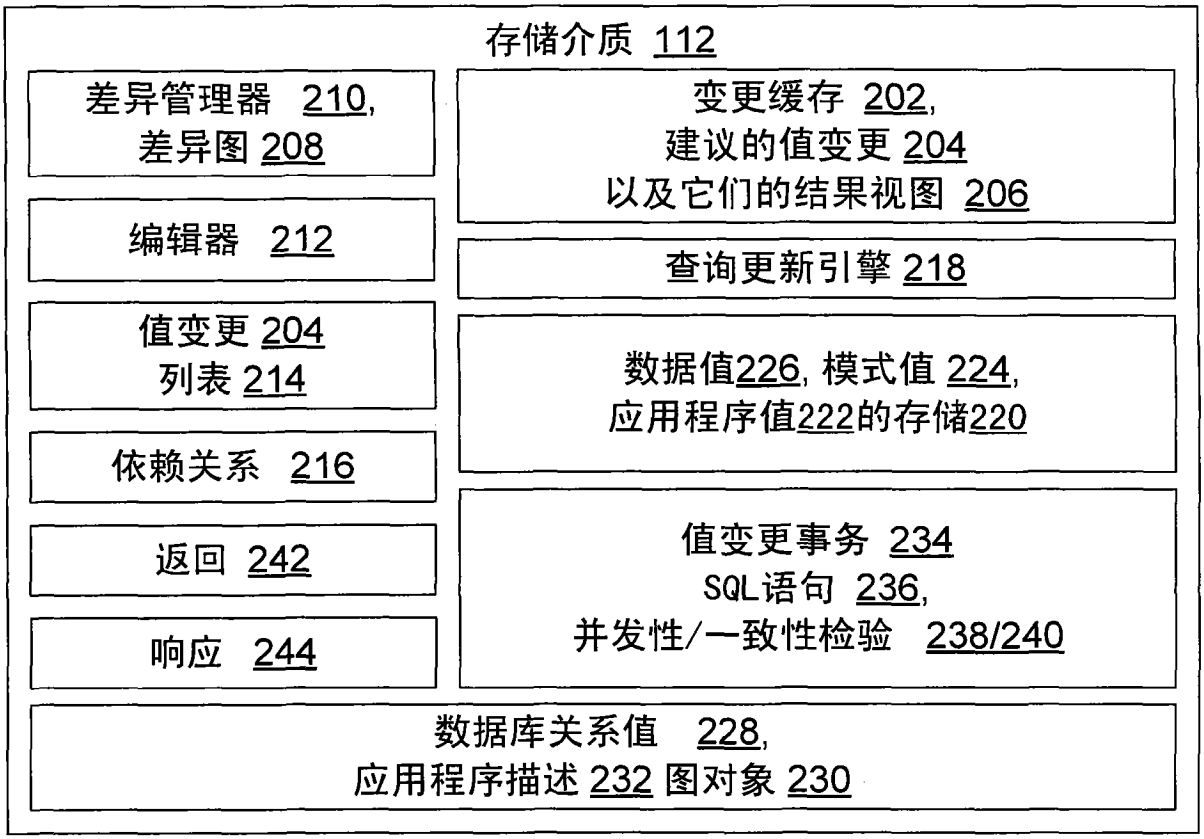

存储介质 112
差异管理器 210,
差异图 208
变更缓存 202,
建议的值变更 204
以及它们的结果视图 206
编辑器 212
查询更新引擎 218
值变更 204
列表 214
数据值226, 模式值 224,
应用程序值222的存储220
依赖关系 216
返回 242
值变更事务 234
SQL语句 236,
并发性/一致性检验 238/240
响应 244
数据库关系值 228,
应用程序描述 232 图对象 230

图 2


300
收集302建议的变更
标识304前提/后置条件306/308
通过判断312来分析310依赖关系
排序/放置314 / 316以形成318变更列表
提交320变更，获取322返回
获取324访问，评估326变更
定位328未满足的条件，引发330错误332，阻止334变更
验证336前提条件
观察338值（跟踪340观察），通过用户姿势344递交342建议的变更
命令346提交348
接收350对提交命令的响应
指定/提供352 / 354关系356
从姿势产生358 差异图
维护360 差异图，产生362视图
访问364多用户存储
产生/递交366/368值变更
检测370指示结果的返回

图 3

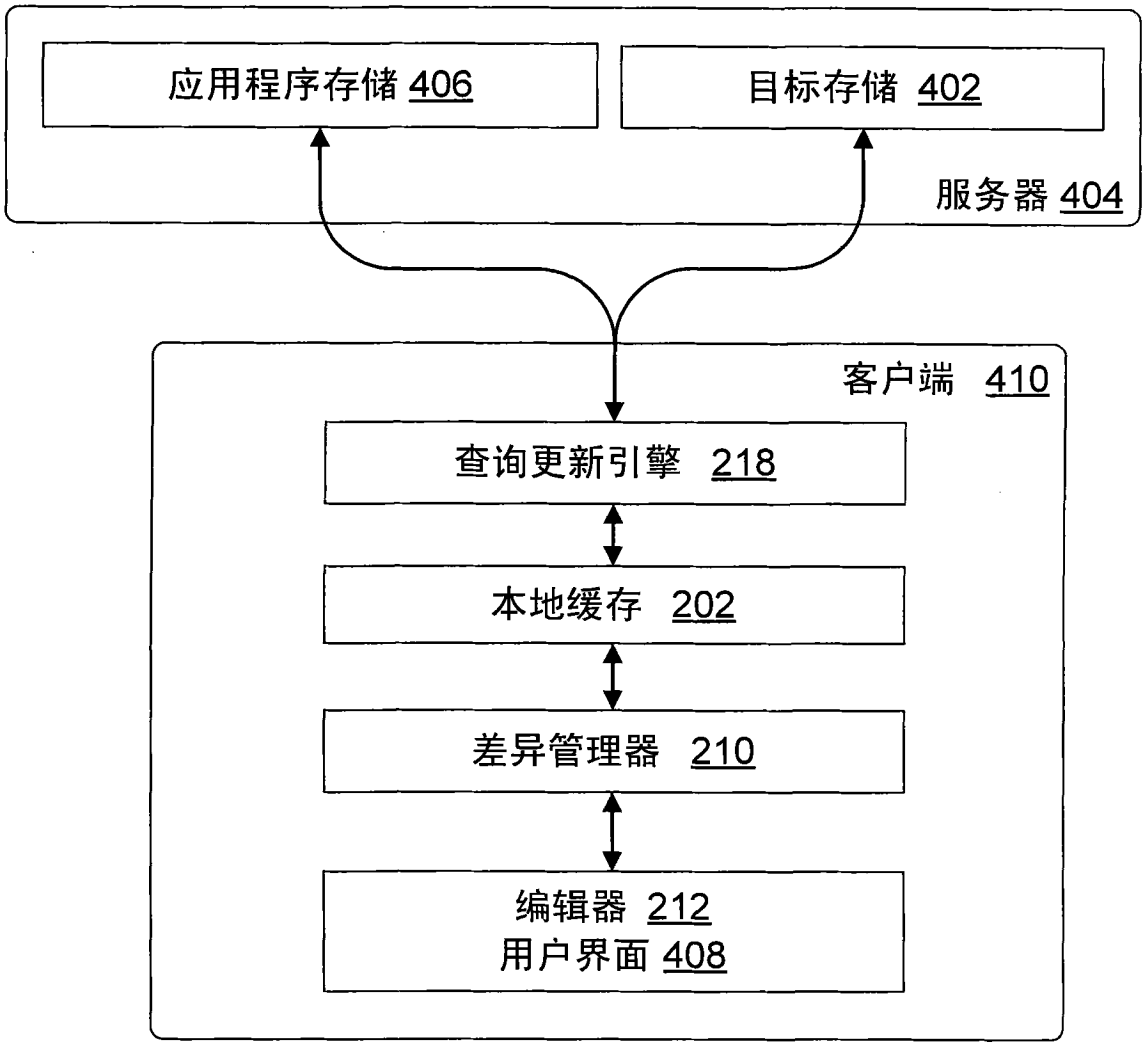

应用程序存储 406
目标存储 402
服务器 404
客户端 410
查询更新引擎 218
本地缓存 202
差异管理器 210
编辑器 212
用户界面 408

图 4