(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116894022 A
(43)申请公布日 2023.10.17

(21)申请号 202310286431.8
(22)申请日 2023.03.22
(30)优先权数据
17/708,440 2022.03.30 US
(71)申请人 国际商业机器公司
地址 美国纽约
(72)发明人 钟嘉田 姜朋慧 刘东慧 沈星星 于佳 殷勇 路京 汤晓燕
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
专利代理师 吴信刚
(51)Int.Cl.
G06F 16/21(2019.01)
G06F 16/23(2019.01)
G06F 11/30(2006.01)
G06F 11/34(2006.01)

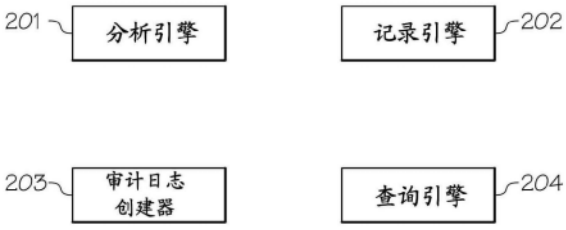
权利要求书2页 说明书21页 附图9页

(54)发明名称

利用结构化审计日志来提高数据库审计的准确性和效率

(57)摘要

本公开涉及利用结构化审计日志来提高数据库审计的准确性和效率。一种用于提高审计数据库的准确性和效率的计算机实现的方法、系统和计算机程序产品。数据库的表、列表或索引被分析以识别元数据，所述元数据包括时间序列数据、用户数据、互联网协议地址和操作数据。所识别的元数据与从中提取元数据的表、列表或索引的对应记录或行相关联。然后，基于对应的数据操作，确定是否记录与分析的表、列表或索引的记录或行相关联的原始数据映像。所识别的元数据以及所记录的数据映像(如果有的话)被存储在结构化审计日志中。然后，基于将与查询相关联的记录或行标识符(RID)和与结构化审计日志相关联的RID进行匹配，从结构化审计日志获得审计信息。



CN 116894022 A

1.一种用于提高数据库审计的准确性和效率的计算机实现的方法,所述方法包括:

分析数据库的表、列表或索引以识别元数据,其中所述元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据;

将所述识别的元数据与所述数据库的所述分析的表、列表或索引的对应记录或行相关联;

基于对应的数据操作确定是否记录与所述数据库的所述分析的表、列表或索引的所述记录或行相关联的数据映像;以及

将所述识别的元数据和记录的数据映像(如果有的话)存储在结构化审计日志中,所述结构化审计日志与所述数据库的所述分析的表、列表或索引的所述记录或行的记录或行标识符相关联。

2.根据权利要求1所述的方法,进一步包括:

当存在并发事务时,记录由读取操作产生的数据的映像;以及

响应于当存在所述并发事务时将由所述读取操作产生的所述数据的所述记录映像存储在所述结构化审计日志中,从所述结构化审计日志获得涉及事务并发性的脏读数据。

3.根据权利要求1所述的方法,进一步包括:

将结构化查询语言数据定义语言语句记录为数据定义语言操作的数据映像;以及

将结构化查询语言表达式记录为批量数据操纵语言操作的数据映像。

4.根据权利要求1所述的方法,进一步包括:

将所述数据库的所述分析的表、列表或索引的所述记录或行的查询链接到所述数据库的所述分析的表、列表或索引的所述记录或行的先前执行的查询。

5.根据权利要求4所述的方法,进一步包括:

在所述结构化审计日志中存储指针,以将所述数据库的所述分析的表、列表或索引的所述记录或行的所述查询链接到所述数据库的所述分析的表、列表或索引的所述记录或行的先前执行的查询。

6.根据权利要求1所述的方法,进一步包括:

接收审计所述数据库的查询请求;以及

结合审计所述数据库所述查询请求识别所述数据库的表、列表或索引的记录或行标识符。

7.根据权利要求6所述的方法,进一步包括:

响应于与所述查询请求相关联的所述识别的记录或行标识符和与所述结构化审计日志相关联的所述记录或行标识符之间的匹配,从所述结构化审计日志中检索经审计的信息。

8.一种用于提高数据库审计的准确性和效率的计算机程序产品,所述计算机程序产品包括具有与其一起实施的程序代码的一个或多个计算机可读存储介质,所述程序代码包括用于实现根据权利要求1至7中的一项所述的步骤的程序指令。

9.一种系统,包括:

存储器,用于存储计算机程序,所述计算机程序用于提高数据库审计的准确性和效率;以及

连接到所述存储器的处理器,其中所述处理器被配置为执行所述计算机程序的程序指

令以实现根据权利要求1至7中的一项所述的步骤。

利用结构化审计日志来提高数据库审计的准确性和效率

技术领域

[0001] 本公开总体上涉及数据库审计，并且更具体地涉及利用结构化审计日志来提高数据库审计的准确性和效率。

背景技术

[0002] 数据库审计是对所选择的用户数据库动作的监视和记录。它可以基于单独的动作，诸如所执行的SQL语句的类型，或诸如用户名、应用、时间等的因素的组合。

[0003] 这种审计典型地用于实现对于在特定模式、表或行中采取的或影响特定内容的当前动作的未来可计量性。此外，这样的审计可以用于基于该可计量性来阻止用户(或其他人)进行不适当的动作。此外，这样的审计使得人们能够调查可疑活动。例如，如果用户正从表中删除数据，那么安全管理员可能决定审计到数据库的所有连接以及从数据库中的所有表中的行的所有成功和不成功删除。另外，这样的审计可用于监视和收集关于特定数据库活动的数据。例如，可收集关于哪些表正被更新、执行了多少逻辑输入/输出操作、或多少并发用户在峰值时间连接的统计。此外，这样的审计可以用于检测授权或访问控制实现的问题。

[0004] 基于数据库审计的此类使用，识别数据库的哪些行和列以及被谁和何时被访问是重要的。不幸的是，这样的信息不容易被跟踪和记录，尤其是对于NoSQL(“非结构化查询语言(SQL)”)数据库。

[0005] NoSQL数据库提供了用于存储和检索数据的机制，该机制是以除在关系数据库中使用的表关系之外的手段建模的。NoSQL数据库使用的数据结构(例如，键-值对)与关系数据库中默认使用的数据结构不同，从而使得一些操作在NoSQL中更快。此外，NoSQL数据库利用非结构化存储，其允许大规模地高性能、灵活的信息处理。例如，NoSQL数据库可以跨多个处理节点以及跨多个服务器存储非结构化数据。然而，通过跨多个处理节点以及跨多个服务器来存储非结构化数据，这使得对这样的数据库的审计成为挑战。

[0006] 目前，审计插件可以用于尝试监视和记录数据库动作，例如用于NoSQL数据库。不幸的是，仅记录数据操作(例如，更新操作、读取操作)。作为监视和记录有限数据的结果，审计在识别数据库的哪些行和列被访问以及被谁和何时被访问方面是有缺陷的。

[0007] 可替代地，时间表可以用于尝试监视和记录数据库动作，诸如用于NoSQL数据库。不幸的是，一些操作未被记录。如在使用审计插件的场景中，时间表的使用导致有限的数据被监视和记录，从而导致审计在识别数据库的哪些行和列以及被谁和何时被访问方面是有缺陷的。此外，时间表的使用是耗时且低效的，尤其涉及对数据库的频繁读取、添加、删除和更新操作。另外，在事务并发性(两个事务一起运行，诸如在重叠的时间段期间访问相同的数据库行)中获得特定信息(诸如脏读数据)的请求可能无法使用时间表来实现。

[0008] 因此，当前不存在用于以实现对数据库(例如，NoSQL数据库)的数据访问的准确证明的高效方式有效地审计数据库动作的手段。

发明内容

[0009] 在本公开的一个实施例中，一种用于提高数据库审计的准确性和效率的计算机实现的方法包括分析数据库的表、列表或索引以识别元数据，其中所述元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据。该方法还包括将所识别的元数据与该数据库的所分析的表、列表或索引的对应记录或行相关联。该方法另外包括基于对应的数据操作来确定是否记录与该数据库的所分析的表、列表或索引的记录或行相关联的数据映像。此外，该方法包括将所识别的元数据和所记录的数据映像(如果有的话)存储在结构化审计日志中，所述结构化审计日志与该数据库的所分析的表、列表或索引的记录或行的记录或行标识符相关联。

[0010] 以这种方式，通过利用结构化审计日志来提高数据库审计的准确性和效率。

[0011] 在本公开的另一个实施例中，一种用于提高数据库审计的准确性和效率的计算机程序产品，其中所述计算机程序产品包括一个或多个计算机可读存储介质，所述一个或多个计算机可读存储介质具有随其实施的程序代码，其中所述程序代码包括用于分析数据库的表、列表或索引以识别元数据的程序指令，其中所述元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据。该程序代码进一步包括用于将所识别的元数据与该数据库的所分析的表、列表或索引的对应记录或行相关联的编程指令。该程序代码另外包括用于基于对应的数据操作来确定是否记录与该数据库的所分析的表、列表或索引的记录或行相关联的数据映像的编程指令。此外，该程序代码包括用于将所识别的元数据和所记录的数据映像(如果有的话)存储在结构化审计日志中的编程指令，所述结构化审计日志与该数据库的所分析的表、列表或索引的记录或行的记录或行标识符相关联。

[0012] 以这种方式，通过利用结构化审计日志提高了数据库审计的准确性和效率。

[0013] 在本公开的又一实施例中，一种系统包括用于存储用于提高数据库审计的准确性和效率的计算机程序的存储器和连接至该存储器的处理器。该处理器被配置为执行计算机程序的程序指令，包括分析数据库的表、列表或索引以识别元数据，其中元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据。该处理器进一步被配置为执行计算机程序的程序指令，包括将识别的元数据与该数据库的所分析的表、列表或索引的对应记录或行相关联。该处理器另外被配置成执行计算机程序的程序指令，包括基于对应的数据操作来确定是否记录与该数据库的所分析的表、列表或索引的记录或行相关联的数据映像。此外，处理器被配置为执行计算机程序的程序指令，包括将所识别的元数据和所记录的数据映像(如果有的话)存储在结构化审计日志中，所述结构化审计日志与该数据库的所分析的表、列表或索引的记录或行的记录或行标识符相关联。

[0014] 以此方式，通过利用结构化审计日志提高了数据库审计的准确性和效率。

[0015] 前述内容已经相当概括地概述了本公开的一个或多个实施例的特征和技术优点，以便可以更好地理解以下本公开的详细描述。本公开的附加特征和优点将在下文中描述，其可形成本公开的权利要求的主题。

附图说明

[0016] 当结合以下附图考虑以下详细描述时，可以获得对本公开的更好理解，其中：

[0017] 图1示出了根据本公开的实施例的用于实施本公开的原理的通信系统；

[0018] 图2是根据本公开的实施例的通过利用结构化审计日志来提高数据库审计的准确性和效率的数据库管理系统的软件部件的图；

[0019] 图3示出了数据库管理系统的硬件配置的本公开实施例，其代表用于实施本公开的硬件环境；

[0020] 图4是根据本公开的实施例的用于建立结构化审计日志的方法的流程图；

[0021] 图5示出了根据本公开的实施例的涉及更新操作的结构化审计日志；

[0022] 图6示出了根据本公开的实施例的涉及读取或查询操作的结构化审计日志；

[0023] 图7示出了根据本公开的实施例的涉及数据定义语言(DDL)操作的结构化审计日志；

[0024] 图8示出了根据本公开的实施例的涉及批量数据操纵语言(DML)操作的结构化审计日志；

[0025] 图9是根据本公开的实施例的通过利用结构化审计日志来提高数据库审计的准确性和效率的方法的流程图；以及

[0026] 图10示出了根据本公开的实施例的将与结构化审计日志相关联的记录或行标识符(RID)与和用于审计数据库的查询相关联的RID进行匹配和不匹配。

具体实施方式

[0027] 如背景技术部分中所述，当前不存在用于以高效方式有效地审计数据库动作的手段，该手段实现对数据库(例如，NoSQL数据库)的数据访问的准确证明。

[0028] 本公开的实施例提供了一种用于建立和使用结构化审计日志用于数据库(例如，NoSQL数据库)的数据访问的准确证明的手段。以下将提供这些和其他特征的更详细的描述。

[0029] 在本公开的一些实施例中，本公开包括用于提高审计数据库的准确性和效率的计算机实现的方法、系统和计算机程序产品。在本公开的一个实施例中，分析数据库的表、列表或索引以识别元数据，所述元数据包括时间序列数据、用户数据、互联网协议地址和操作数据。如本文中所使用的“元数据”是指提供关于其他数据的信息的数据。如在本文中所使用的“时间序列数据”是指发出查询以对来自数据库的表、列表或索引的信息进行更新、插入、删除等的特定顺序。如本文所使用的“用户数据”是指发出查询以访问数据库的表、列表或索引的记录或行的特定用户的标识符。如在此所使用的“互联网协议(IP)地址”是指分配给设备(如数据库)的唯一标识符。如本文所使用的“操作数据”是指通过由计算设备的用户发出的查询而请求在数据库上执行的操作。在一个实施例中，这样的操作包括更新操作、删除操作、读取操作、数据定义语言操作(用于定义数据结构，例如结构化查询语言中的创建表、更改表、删除表、创建视图等)、批量数据操纵语言操作(用于操纵数据库的表、列表或索引的多个记录或行)等。所识别的元数据与从中提取元数据的数据库的表、列表或索引的对应记录或行相关联。然后，基于对应的数据操作，确定是否记录与所分析的该数据库的表、列表或索引的记录或行相关联的原始数据映像。如本文中所使用的“原始数据映像”或“数据映像”是指由数据操作(例如，更新操作)产生的数据的映像。例如，可以响应于更新或插入操作来记录从数据操作得到的数据的数据映像。然而，除存在并发事务之外，可能不响应于删除操作或响应于读取/查询操作来记录从数据操作得到的数据的数据映像。然后，将所

识别的元数据以及所记录的数据映像(如果有的话)存储在“结构化审计日志”中,该“结构化审计日志”与在其上获得这种结构化信息的数据库的分析的表、列表或索引的记录或行的记录或行标识符相关联。如本文所使用的“结构化审计日志”是指存储所述审计信息(例如,元数据、记录的数据映像等)的数据结构。在一个实施例中,结构化审计日志被存储在数据库管理系统的存储设备(例如,存储器、磁盘单元)中,其中审计数据库的未来查询可以访问这样的存储的结构化审计日志以获得关于数据库的审计信息。在一个实施例中,基于将与查询相关联的行或记录标识符同与包含所请求的数据库审计信息的结构化审计日志相关联的记录或行标识符匹配,从所存储的结构化审计日志获得适当的审计信息。以这种方式,通过利用结构化审计日志来提高数据库审计的准确性和效率。

[0030] 在以下描述中,阐述了许多具体细节以便提供对本公开的透彻理解。然而,对本领域的技术人员而言将显而易见的是,可以在没有此类具体细节的情况下实践本公开。在其他实例中,以框图的形式示出了众所周知的电路,以便不以不必要的细节模糊本公开。对于大部分,已经省略考虑时序考虑等的细节,因为这样的细节对于获得对本公开的完整理解不是必需的,并且在本领域的普通技术人员的技能之内。

[0031] 现在详细参照附图,图1示出了用于实施本公开的原理的通信系统100的本公开的实施例。通信系统100包括经由网络103连接到数据库管理系统102的计算设备101。此外,如图1所示,数据库管理系统102连接到数据库104(例如,NoSQL数据库)。

[0032] 计算设备101可以是配置有连接至网络103并且因此与其他计算设备101和数据库管理系统102通信的能力的任何类型的计算设备(例如,便携式计算单元、个人数字助理(PDA)、膝上型计算机、移动设备、平板个人计算机、智能电话、移动电话、导航设备、游戏单元、台式计算机系统、工作站、互联网电器等)。注意,计算设备101和计算设备101的用户两者都可用元素号101来标识。

[0033] 网络103可以是例如局域网、广域网、无线广域网、电路交换电话网、全球移动通信系统(GSM)网络、无线应用协议(WAP)网络、WiFi网络、IEEE 802.11标准网络、其各种组合等。在不脱离本公开范围的情况下,其他网络(为简洁起见,在此省略其描述)也可结合图1的系统100使用。

[0034] 在一个实施例中,计算设备101的用户向数据库管理系统102发出查询(例如,结构化查询语言(SQL)查询、查询JSON(JavaScript®对象表示法)对象)以对来自数据库104的信息进行更新、插入、删除等。例如,用户可以发出INSERT INTO查询,以向数据库104中的表添加新的数据行。这样的查询将由数据库管理系统102处理,诸如存储和检索用户所请求的数据。

[0035] 在一个实施例中,数据库管理系统102被配置成维护数据库104,如关系数据库或NoSQL数据库。在一个实施例中,数据库管理系统102对应于SQL服务器,该SQL服务器被配置为使用结构化查询语言(SQL)来查询和维护数据库104。在一个实施例中,数据库管理系统102对应于被配置为查询存储在NoSQL数据库104中的JSON(JavaScript®对象表示法)数据(例如,JSON对象)的服务器。

[0036] 在一个实施例中,如以下进一步讨论的,数据库管理系统102被配置成构建并使用结构化审计日志用于数据库104的数据访问的准确证明。在一个实施例中,如果适用,这样的结构化审计日志包括元数据(例如,时间序列数据、用户数据、互联网协议(IP)地址和操

作数据)以及原始数据映像。

[0037] 如在此使用的“时间序列数据”是指查询被发出以对来自数据库104的表、列表或索引的信息进行更新、插入、删除等的特定顺序。例如,访问一行的第一查询可具有TS1的时间序列数据(时间序列#1),并且访问同一行的随后查询可具有TS2的时间序列数据,从而表示该查询在与TS1相关联的查询之后。

[0038] 如在此所使用的“用户数据”是指发出查询以访问数据库104的表、列表或索引的记录或行的特定用户的标识符。

[0039] 如在此使用的“互联网协议(IP)地址”是指分配给设备(如数据库104)的唯一标识符。在一个实施例中,包括在元数据中的IP地址对应于由计算设备101的用户正在查询的数据库104的IP地址。

[0040] 如在此使用的“操作数据”是指正由计算设备101的用户发出的查询请求在数据库104上执行的操作。在一个实施例中,这样的操作包括更新操作、删除操作、读取操作、数据定义语言操作(用于定义数据结构,例如结构化查询语言中的创建表、更改表、删除表、创建视图等)、批量数据操纵语言操作(用于操纵数据库104的表、列表或索引的多个记录或行)等。

[0041] 如在此使用的“原始数据映像”是指由数据操作(例如,更新操作)产生的数据的映像。在一个实施例中,这样的映像可仅包括改变的列或行的数据。

[0042] 在一个实施例中,这种信息(例如,元数据、原始数据映像)与数据库104的表、列表或索引的记录或行标识符相关联,其中这种标识符标识在其上执行数据操作的记录或行。

[0043] 在一个实施例中,这种信息(例如,元数据、原始数据映像)连同数据库104的表、列表或索引的记录或行标识符一起被数据库管理系统102存储在“结构化审计日志”中。如本文所使用的“结构化审计日志”是指存储所述信息的数据结构。

[0044] 在一个实施例中,结构化审计日志进一步包括指向先前链接的查询的指针,该先前链接的查询在与所讨论的元数据和原始数据映像相关联的查询(如果适用的话)之前被执行。

[0045] 在一个实施例中,结构化审计日志被存储在数据库管理系统102的存储设备(例如,存储器、磁盘单元)中。

[0046] 在建立结构化审计日志后,数据库管理系统102使用结构化审计日志用于数据库104的数据访问的准确证明。

[0047] 以下将进一步提供这些和其他特征的更详细的描述。此外,下文结合图2提供数据库管理系统102的软件组件的描述,且下文结合图3进一步提供数据库管理系统102的硬件配置的描述。

[0048] 系统100的范围不限于任何一个特定网络架构。系统100可以包括任意数量的计算设备101、数据库管理系统102、网络103和数据库104。

[0049] 以下结合图2提供关于由数据库管理系统102用于通过利用结构化审计日志来提高数据库审计的准确性和效率的软件组件的讨论。

[0050] 图2是根据本公开的实施例的通过利用结构化审计日志来提高数据库审计的准确性和效率的数据库管理系统102的软件组件的图。

[0051] 结合图1参见图2,数据库管理系统102包括分析引擎201,该分析引擎201被配置成

分析数据库104的表、列表和/或索引以标识元数据，该元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据。

[0052] 如在此使用的“表(table)”是指以表格式保持在数据库104内的相关数据的集合。

[0053] 如在此使用的“列表(list)”是指一组条目或值，诸如存储在数据库104的字段中的条目或值。

[0054] 如在此使用的“索引”是指提高对数据库104的数据库表的数据检索操作的速度的数据结构。索引用于快速定位数据，而不必在每次访问数据库表时搜索数据库表中的每个行。在一个实施例中，使用数据库表的一个或多个列来创建索引，从而为有序记录的高效访问和快速随机查找提供基础。在一个实施例中，索引是从表中选择的数据列的副本，其被设计为实现高效搜索。在一个实施例中，索引包括“键”或到从其复制了索引的原始数据行的直接链接，以允许高效地检索完整的行。

[0055] 如在此使用的“元数据”是指提供关于其他数据的信息的数据。在一个实施例中，元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据。

[0056] 如上所述，如在本文中使用的“时间序列数据”是指查询被发出以对来自数据库104的表、列表或索引的信息进行更新、插入、删除等的具体顺序。例如，访问一行的第一查询可具有TS1的时间序列数据(时间序列#1)，并且访问同一行的随后查询可具有TS2的时间序列数据，从而指示该查询在与TS1相关联的查询之后。

[0057] 如在此所使用的“用户数据”是指发出查询以访问数据库104的表、列表或索引的记录或行的特定用户的标识符。

[0058] 如在此使用的“互联网协议(IP)地址”是指分配给设备(如数据库104)的唯一标识符。在一个实施例中，包括在元数据中的IP地址对应于由计算设备101的用户正在查询的数据库104的IP地址。

[0059] 如在此使用的“操作数据”是指正由计算设备101的用户发出的查询请求在数据库104上执行的操作。在一个实施例中，这样的操作包括更新操作、删除操作、读取操作、数据定义语言操作(用于定义数据结构，例如结构化查询语言中的创建表、更改表、删除表、创建视图等)、批量数据操纵语言操作(用于操纵数据库104的表、列表或索引的多个记录或行)等。

[0060] 在一个实施例中，分析引擎201经由方法调用(诸如使用连接对象的getMetaData()方法)从数据库104的表、列表或索引中识别元数据。

[0061] 在一个实施例中，分析引擎201利用数据库文件编制工具从数据库104的表、列表或索引中提取元数据。从数据库104的表、列表或索引提取元数据的数据库文件编制工具的示例包括但不限于dbdocs.io、Dataedo、Apex®SQL RedGate® SQL、SchemaSpy、dbForge Documente、DBScribe、SentryOne®DOC xPress、Innovasys DocumentX等。

[0062] 分析引擎201还被配置为将识别的元数据与从中提取元数据的表、列表和/或索引的对应记录或行相关联。在一个实施例中，这种关联是通过记录或行标识符(“RID”)，其标识表、列表或索引的特定记录或行。在一个实施例中，记录或行标识符(“RID”)对应于表、列表或索引中的行的地址。

[0063] 例如，如果元数据是从对应于234567的记录或行标识符(“RID”)的数据库104的表的记录或行中提取的，那么从这样的记录或行中提取的元数据与RID 234567相关联。

[0064] 在一个实施例中，分析引擎201经由诸如使用SQL的ROWID语句来识别RID。

[0065] 在一个实施例中，分析引擎201还被配置为基于时间序列数据，将数据库104的分析的表、列表或索引的记录或行的查询与先前执行的查询(如果有的话)链接。例如，由计算设备101的用户发出的请求从数据库104的表、列表或索引的特定记录或行访问数据(诸如操纵或检索数据)的查询可以发生在时间T3。访问数据库104的表、列表或索引的相同记录或行的在先查询可以发生在时间T2，该时间T2可以在查询在时间T1访问数据库104的表、列表或索引的相同记录或行之后发生。在时间T3发生的查询然后可以被链接到在时间T2和T1发生的查询。

[0066] 在一个实施例中，这样的链接可以经由由分析引擎201提供的指针来完成，所述指针诸如与在时间T3的查询相关联的结构化审计日志中的指针，所述指针指向与在时间T2的查询相关联的结构化审计日志，该结构化审计日志包含指向与在时间T1的查询相关联的结构化审计日志的指针。以这种方式，可以容易地检索历史查询的结果集(如果适用，具有元数据和数据映像的结构化审计日志)。

[0067] 在一个实施例中，分析引擎201基于与查询相关联的记录或行标识符(“RID”)确定这种链接。例如，在时间T1、T2和T3发生的查询可以全部与相同的RID(诸如12345)相关联。在一个实施例中，分析引擎201例如在一数据结构(例如，表)中存储数据库104的所分析的表、列表或索引的RID。在一个实施例中，分析引擎201还将与RID相关联的时间序列数据存储在该数据结构中。这样的时间序列数据可以用于链接与过去发生的与相同RID相关联的查询相关联的结构化审计日志(包含如下文进一步讨论的时间序列数据)。在一个实施例中，这样的数据结构被存储在数据库管理系统102的存储设备(例如，存储器、磁盘单元)中。

[0068] 数据库管理系统201进一步包括记录引擎202，所述记录引擎202被配置成记录由数据操作(例如，更新操作)产生的数据的原始数据映像，诸如由对数据库104的表的特定行执行的更新操作产生的数据。如在此使用的“原始数据映像”是指由数据操作(例如，更新操作)产生的数据的映像。

[0069] 在一个实施例中，记录引擎202基于数据操作的类型来确定是否记录从数据操作得到的数据的数据映像。

[0070] 在一个实施例中，记录引擎202从所提取的元数据中确定操作类型(例如，更新、删除、插入、读取、数据定义语言、批量数据操纵语言等)。例如，分析引擎201可能已经从表的行(用RID 12345标识)中提取了元数据，该元数据具有时间序列(TS)数据TS3、用户数据“用户1”、数据库104的IP地址“IP1”、以及表示更新操作的数据操作“UPD”。

[0071] 如以上所讨论的，在一个实施例中，记录引擎202基于对应的数据操作确定是否记录与数据库104的所分析的表、列表或索引的记录或行相关联的原始数据映像。在一个实施例中，记录引擎202在执行所述操作之后记录由更新(“UPD”)或插入(“INS”)操作产生的数据的数据映像。在一个实施例中，除了当存在并发事务时，记录引擎202不记录由读取/查询操作(“QRY”)产生的数据的数据映像。以这种方式，可以获得涉及事务并发性的脏读数据，如下面进一步讨论的。如本文中所使用的“脏读”在事务读取尚未提交的数据时发生。例如，假设事务1更新数据库104的一个行。事务2在事务1提交该更新之前读取更新的行。这种情况被认为对应于“脏读”。

[0072] 在一个实施例中，记录引擎202不记录由删除操作(“DLT”)产生的数据的数据映

像。

[0073] 在一个实施例中，记录引擎202将SQL数据定义语言(DDL)语句记录为由DDL操作产生的数据映像。在一个实施例中，“DDL”操作用于定义数据结构，诸如结构化查询语言(SQL)中的创建表、更改表、删除表、创建视图等。

[0074] 在一个实施例中，记录引擎202将SQL表达式记录为由批量数据操纵语言(DML)操作产生的数据映像。在一个实施例中，DML操作用于操纵数据本身。DML操作的示例包括使用SQL的插入、更新或删除指令。

[0075] 如果适用，由记录引擎202用来记录数据映像的软件工具的示例包括但不限于Equalum®、Hevo Data、HVR、IBM® WebSphere®、Qlik®、Oracle® GoldenGate®、Precisely®、Striim®、Talend®等。

[0076] 数据库管理系统102进一步包括审计日志创建器203，该审计日志创建器被配置成基于从分析引擎201和记录引擎202获得的信息来创建结构化审计日志。如本文所使用的“结构化审计日志”是指存储由分析引擎201提取和识别的元数据、由记录引擎202记录的原始数据映像(如果有的话)、以及由分析引擎201提供的指向与历史查询相关联的结构化审计日志的任何指针的数据结构。在一个实施例中，此类结构化审计日志被存储在数据库管理系统102的存储设备(例如，存储器、磁盘单元)中。

[0077] 在一个实施例中，审计日志创建器203将由分析引擎201提取和识别的元数据、由记录引擎202记录的原始数据映像(如果有的话)、以及由分析引擎201提供的指向与历史查询相关联的结构化审计日志的任何指针存储在所创建的结构化审计日志中。

[0078] 由审计日志创建器203用来创建结构化审计日志并且将以上讨论的信息存储在这种创建的结构化审计日志中的软件工具的示例包括但不限于erwin®Data Modeler、ER/Studio®、DbSchema、ERBuilder、HeidiSQL、Navicat®Data Modeler、Toad Data Modeler、Archi等。

[0079] 数据库管理系统102另外包括查询引擎204，该查询引擎204被配置为接收并处理从计算设备101的用户发出的查询，包括用于经由结构化审计日志来审计数据库104的查询。

[0080] 在一个实施例中，查询引擎204(例如，Presto®、Apache®Drill、Cloudera® Impala、Apache®Spark等)被配置成从计算设备101的用户接收用于审计数据库104的查询请求。在一个实施例中，用于审计数据库104的查询包括对其执行审计的数据库104的表、列表或索引的记录或行的记录或行标识符(RID)。在一个实施例中，查询引擎204搜索与结构化审计日志相关联的RID以确定是否存在匹配。在一个实施例中，每个结构化审计日志与一RID相关联，其中这样的信息被存储在数据结构中。在一个实施例中，查询引擎204在这样的数据结构中搜索以识别任何匹配的RID。在一个实施例中，查询引擎204利用自然语言处理来识别数据结构中的任何匹配。在一个实施例中，这样的数据结构被存储在数据库管理系统102的存储设备(例如，存储器、磁盘单元)中。

[0081] 在一个实施例中，查询引擎204分析来自计算设备101的用户的用于审计数据库104的查询请求，以确定数据库104的表、列表或索引的哪些记录或行要进行审计。例如，查询可以包括记录或行号函数或选择子句，其识别数据库104的表、列表或索引的记录或行以

进行审计。在识别数据库104的表、列表或索引的记录或行时，可以经由ROWID语句来识别记录或行标识符(RID)。

[0082] 一旦识别了RID，查询引擎204可以确定在这样的RID和存储在上述数据结构中的结构化审计日志的RID之间是否存在匹配。例如，如以上所讨论的，在一个实施例中，查询引擎204利用自然语言处理来识别数据结构中的任何匹配。

[0083] 如果在与查询相关联的RID和结构化审计日志的RID之间存在匹配，则查询引擎204从匹配的结构化审计日志检索经审计的信息(例如，元数据、原始数据映像)。然后，诸如经由计算设备101的用户界面，将这样的经审计的信息提供给计算设备101的用户。

[0084] 可替代地，如果在与查询相关联的RID和结构化审计日志的RID之间不存在匹配，则查询引擎204向请求者(计算设备101的用户)通知，数据库104的表、列表或索引的该记录或行尚未被访问。

[0085] 下文结合对通过利用结构化审计日志来提高数据库审计的准确性和效率的方法的讨论提供了对这些和其他功能的进一步描述。

[0086] 在讨论通过利用结构化审计日志来提高数据库审计的准确性和效率的方法之前，以下结合图3提供数据库管理系统102(图1)的硬件配置的描述。

[0087] 现在参见图3，图3示出了代表用于实施本公开的硬件环境的、数据库管理系统102(图1)的硬件配置的本公开的实施例。

[0088] 数据库管理系统102具有通过系统总线302连接至不同其他部件的处理器301。操作系统303在处理器301上运行并提供对图3的不同组件的功能的控制和协调。根据本公开的原理的应用304结合操作系统303运行，并且提供对操作系统303的调用，其中所述调用实现要由应用304执行的不同功能或服务。应用304可包括例如分析引擎201(图2)、记录引擎202(图2)、审计日志创建器203(图2)和查询引擎204(图2)。此外，应用304可以包括例如用于通过利用结构化审计日志来提高数据库审计的准确性和效率的程序，如以下结合图4-10进一步讨论的。

[0089] 再次参见图3，只读存储器(“ROM”)305连接至系统总线302并且包括控制数据库管理系统102的某些基本功能的基本输入/输出系统(“BIOS”)。随机存取存储器(“RAM”)306和盘适配器307也连接到系统总线302。应注意，包括操作系统303和应用304的软件组件可加载到RAM 306中以供执行，RAM 306可为数据库管理系统102的主存储器。盘适配器307可以是与盘单元308(例如，盘驱动器)通信的集成驱动电子器件(“IDE”)适配器。注意，如以下结合图4-10进一步讨论的，用于通过利用结构化审计日志来提高数据库审计的准确性和效率的程序可以驻留在盘单元308中或应用304中。

[0090] 数据库管理系统102可以进一步包括连接至总线302的通信适配器309。通信适配器309将总线302与外部网络(例如，图1的网络103)互连以与诸如计算设备101(图1)的其他设备通信。

[0091] 在一个实施例中，数据库管理系统102的应用304包括分析引擎201、记录引擎202、审计日志创建器203和查询引擎204的软件组件。在一个实施例中，这样的组件可以以硬件实现，其中这样的硬件组件将连接到总线302。以上讨论的由这些组件执行的功能不是通用计算机功能。因此，数据库管理系统102是作为实现特定的、非通用的计算机功能的结果的特定机器。

[0092] 在一个实施例中，数据库管理系统102的此类软件组件(例如，分析引擎201、记录引擎202、审计日志创建器203和查询引擎204)的功能包括用于通过利用结构化审计日志来提高数据库审计的准确性和效率的功能，其可以实施在专用集成电路中。

[0093] 本发明可以是在整合的任何可能的技术细节层次的系统、方法、和/或计算机程序产品。计算机程序产品可包括其上具有用于使处理器执行本发明的各方面的计算机可读程序指令的计算机可读存储介质(或多个介质)。

[0094] 计算机可读存储介质可以是能够保留和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质可以是，例如但不限于，电子存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备、或者上述的任意合适的组合。计算机可读存储介质的更具体示例的非穷尽列表包括以下各项：便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式紧凑盘只读存储器(CD-ROM)、数字通用盘(DVD)、记忆棒、软盘、诸如穿孔卡之类的机械编码设备或具有记录在其上的指令的槽中的凸出结构、以及上述各项的任何合适的组合。如本文所使用的计算机可读存储媒体不应被解释为暂时性信号本身，例如无线电波或其他自由传播的电磁波、通过波导或其他传输媒体传播的电磁波(例如，穿过光纤电缆的光脉冲)或通过电线发射的电信号。

[0095] 在此所描述的计算机可读程序指令可以经由网络(例如，互联网、局域网、广域网和/或无线网络)从计算机可读存储介质下载至对应的计算/处理设备或下载至外部计算机或外部存储设备。网络可以包括铜传输电缆、光传输纤维、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配器卡或网络接口接收来自网络的计算机可读程序指令，并转发计算机可读程序指令以存储在相应计算/处理设备内的计算机可读存储介质中。

[0096] 用于执行本发明的操作的计算机可读程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、集成电路的配置数据、或以一种或多种程序设计语言的任何组合编写的源代码或目标代码，这些程序设计语言包括面向对象的程序设计语言(诸如Smalltalk、C++等)和过程程序设计语言(诸如“C”程序设计语言或类似程序设计语言)。计算机可读程序指令可以完全地在用户计算机上执行、部分在用户计算机上执行、作为独立软件包执行、部分在用户计算机上部分在远程计算机上执行或者完全在远程计算机或服务器上执行。在后一种情况下，远程计算机可通过任何类型的网络(包括局域网(LAN)或广域网(WAN))连接至用户计算机，或者可连接至外部计算机(例如，使用互联网服务提供商通过互联网)。在一些实施例中，包括例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA)的电子电路可以通过利用计算机可读程序指令的状态信息来使电子电路个性化来执行计算机可读程序指令，以便执行本发明的各方面。

[0097] 在此参照根据本发明的实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述本发明的多个方面。应当理解，流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合，都可以由计算机可读程序指令实现。

[0098] 这些计算机可读程序指令可以被提供给计算机的处理器、或其他可编程数据处理装置以便产生机器，这样使得经由计算机的处理器或其他可编程数据处理装置执行的这些

指令创建用于实现在流程图和/或框图的或多个框中指定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中，这些指令使得计算机、可编程数据处理装置、和/或其他设备以特定方式工作，从而，其中存储有指令的计算机可读存储介质包括包含实现流程图和/或框图中的或多个方框中规定的功能/动作的方面的指令的制造品。

[0099] 这些计算机可读程序指令还可以被加载到计算机、其他可编程数据处理装置、或其他设备上，以便使得在该计算机、其他可编程装置或其他设备上执行一系列操作步骤以产生计算机实现的过程，从而使得在该计算机、其他可编程装置、或其他设备上执行的指令实现流程图和/或框图的或多个框中所指定的功能/动作。

[0100] 附图中的流程图和框图示出了根据本发明的不同实施例的系统、方法和计算机程序产品的可能实现方式的架构、功能和操作。对此，流程图或框图中的每个框可表示指令的模块、段或部分，其包括用于实现指定的逻辑功能的一个或多个可执行指令。在一些备选实现中，框中标注的功能可以不按照图中标注的顺序发生。例如，连续示出的两个方框实际上可以作为一个步骤完成，同时、基本上同时、以部分或完全时间上重叠的方式执行，或者方框有时可以以相反的顺序执行，这取决于所涉及的功能。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作或执行专用硬件与计算机指令的组合的专用的基于硬件的系统来实现。

[0101] 如上所述，数据库审计是对所选择的用户数据库动作的监视和记录。它可以基于单独的动作，诸如所执行的SQL语句的类型，或诸如用户名、应用、时间等因素的组合。基于数据库审计的使用，识别数据库的哪些行和列以及被谁和何时被访问是重要的。不幸的是，这样的信息不容易被跟踪和记录，尤其是对于NoSQL(“非结构化查询语言(SQL)”)数据库。NoSQL数据库提供用于数据的存储和检索的机制，该机制以不同于在关系数据库中使用的表关系的手段被建模。NoSQL数据库使用的数据结构(例如，键-值对)与关系数据库中默认使用的数据结构不同，从而使得一些操作在NoSQL中更快。此外，NoSQL数据库利用非结构化存储，其允许大规模地高性能、灵活的信息处理。例如，NoSQL数据库可以跨多个处理节点以及跨多个服务器存储非结构化数据。然而，通过跨多个处理节点以及跨多个服务器来存储非结构化数据，这使得对这样的数据库的审计成为挑战。当前，可以利用审计插件来尝试监视和记录数据库动作，诸如用于NoSQL数据库。不幸的是，仅数据操作(例如，更新操作、读取操作)被记录。作为监控和记录有限数据的结果，审计在识别数据库的哪些行和列被访问以及由谁和何时被访问方面是有缺陷的。可替代地，时间表可以用于尝试监视和记录数据库动作，诸如用于NoSQL数据库。不幸的是，一些操作未被记录。如在使用审计插件的场景中，时间表的使用导致有限数据被监视和记录，从而导致审计在识别数据库的哪些行和列以及由谁和何时被访问方面是有缺陷的。此外，时间表的使用是耗时且低效的，尤其涉及对数据库的频繁读取、添加、删除和更新操作。另外，在事务并发性(两个事务一起运行，诸如在重叠的时间段期间访问相同的数据库行)中获得特定信息(诸如脏读数据)的请求可能无法使用时间表来实现。结果，当前不存在用于以高效方式有效地审计数据库动作的手段，该手段能够实现对数据库(例如，NoSQL数据库)的数据访问的准确证明。

[0102] 本公开的实施例提供了一种用于构建和使用结构化审计日志用于数据库(例如，NoSQL数据库)的数据访问的准确证明的手段，如下文结合图4-10所讨论的。图4是用于建立

结构化审计日志的方法的流程图。图5示出了涉及更新操作的结构化审计日志。图6示出了涉及读取或查询操作的结构化审计日志。图7示出了涉及数据定义语言(DDL)操作的结构化审计日志。图8示出了涉及批量数据操纵语言(DML)操作的结构化审计日志。图9是用于通过利用结构化审计日志来提高数据库审计的准确性和效率的方法的流程图。图10示出了将与结构化审计日志相关联的记录或行标识符(RID)同与审计数据库的查询相关联的RID匹配和不匹配。

[0103] 如上所述,图4是根据本公开的实施例的用于建立结构化审计日志的方法400的流程图。

[0104] 结合图1-3参见图4,在操作401中,数据库管理系统102的分析引擎201分析数据库104的表、列表或索引以识别元数据,该元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据。

[0105] 如以上所讨论的,如在此使用的“表”是指以表格式保存在数据库104内的相关数据的集合。

[0106] 如在此使用的“列表”是指一组条目或值,诸如存储在数据库104的字段中的条目或值。

[0107] 如在此使用的“索引”是指提高对数据库104的数据库表的数据检索操作的速度的数据结构。索引用于快速定位数据,而不必在每次访问数据库表时搜索数据库表中的每个行。在一个实施例中,使用数据库表的一个或多个列来创建索引,从而为有序记录的高效访问和快速随机查找提供基础。在一个实施例中,索引是从表中选择的数据列的副本,其被设计为实现高效搜索。在一个实施例中,索引包括“键”或到从其复制索引的原始数据行的直接链接,以允许高效地检索完整的行。

[0108] 如在此使用的“元数据”是指提供关于其他数据的信息的数据。在一个实施例中,元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据。

[0109] 此外,如上所述,如在本文中使用的“时间序列数据”是指查询被发出以对来自数据库104的表、列表或索引的信息进行更新、插入、删除等的特定顺序。例如,访问一行的第一查询可具有TS1的时间序列数据(时间序列#1),并且访问同一行的随后查询可具有TS2的时间序列数据,从而指示该查询在与TS1相关联的查询之后。

[0110] 如在此所使用的“用户数据”是指发出查询以访问数据库104的表、列表或索引的记录或行的特定用户的标识符。

[0111] 如在此使用的“互联网协议(IP)地址”是指分配给设备(如数据库104)的唯一标识符。在一个实施例中,包括在元数据中的IP地址对应于由计算设备101的用户正在查询的数据库104的IP地址。

[0112] 如在此使用的“操作数据”是指正由计算设备101的用户发出的查询请求在数据库104上执行的操作。在一个实施例中,这样的操作包括更新操作、删除操作、读取操作、数据定义语言操作(用于定义数据结构,例如结构化查询语言中的创建表、更改表、删除表、创建视图等)、批量数据操纵语言操作(用于操纵数据库104的表、列表或索引的多个记录或行)等。

[0113] 在一个实施例中,分析引擎201经由方法调用(诸如使用连接对象的getMetaData()方法)从数据库104的表、列表或索引中识别元数据。

[0114] 在一个实施例中，分析引擎201利用数据库文件编制工具从数据库104的表、列表或索引中提取元数据。从数据库104的表、列表或索引提取元数据的数据库文件编制工具的示例包括但不限于dbdocs.io、Dataedo、Apex®SQL RedGate® SQL、SchemaSpy、dbForge Documente、DBScribe、SentryOne®DOC xPress、Innovasys DocumentX等。

[0115] 在操作402中，数据库管理系统102的分析引擎201将所识别的元数据与从中提取元数据的表、列表或索引的对应记录或行相关联。

[0116] 如以上所讨论的，在一个实施例中，这种关联是通过记录或行标识符(“RID”)，其标识表、列表或索引的特定记录或行。在一个实施例中，记录或行标识符(“RID”)对应于行在表、列表或索引中的地址。

[0117] 例如，如果元数据是从数据库104的表的对应于234567的记录或行标识符(“RID”)的记录或行中提取的，那么从这样的记录或行中提取的元数据与RID 234567相关联。

[0118] 在一个实施例中，分析引擎201经由诸如使用SQL的ROWID语句来识别RID。

[0119] 在操作403中，数据库管理系统102的记录引擎202从所识别的元数据确定数据操作的类型(例如，更新、删除、插入、读取、数据定义语言、数据操纵语言等)。例如，分析引擎201可能已经从表的行(用RID 12345标识)中提取了元数据，该元数据具有时间序列(TS)数据TS3、用户数据“用户1”、数据库104的IP地址“IP1”、以及表示更新操作的数据操作“UPD”。

[0120] 在操作404中，数据库管理系统102的记录引擎202基于对应的数据操作确定是否记录与分析的表、列表或索引的记录或行相关联的原始数据映像。

[0121] 如果记录引擎202基于对应的数据操作确定不记录与分析的表、列表或索引的记录或行相关联的原始数据映像，则在操作405中，数据库管理系统102的记录引擎202不记录与分析的表、列表或索引的记录或行相关联的数据映像。下面进一步提供操作405的进一步解释。

[0122] 然而，如果记录引擎202基于对应的数据操作确定记录与分析的表、列表或索引的记录或行相关联的原始数据映像，则在操作406中，数据库管理系统102的记录引擎202记录由所述操作得到的数据的数据映像，如下文结合图5-8进一步讨论的。

[0123] 如以上所讨论的，如在此所使用的“原始数据映像”或“数据映像”是指从数据操作(例如，更新操作)得到的数据的映像。

[0124] 在一个实施例中，记录引擎202基于数据操作的类型来确定是否记录由数据操作产生的数据的数据映像。

[0125] 在一个实施例中，记录引擎202从所提取的元数据确定操作类型(例如，更新、删除、插入、读取、数据定义语言、批量数据操纵语言等)。例如，分析引擎201可能已经从数据库104的表的行(由RID 12345标识)中提取了元数据，该元数据具有时间序列(TS)数据TS3、用户数据“用户1”、的数据库104的IP地址“IP1”、以及表示更新操作的数据操作“UPD”。

[0126] 如上所述，在一个实施例中，记录引擎202基于对应的数据操作确定是否记录与分析的数据库104的表、列表或索引的记录或行相关联的原始数据映像。在一个实施例中，记录引擎202在执行所述操作之后记录由更新(“UPD”)或插入(“INS”)操作产生的数据的数据映像。在一个实施例中，除了当存在并发事务时，记录引擎202不记录由读取/查询操作(“QRY”)产生的数据的数据映像。以这种方式，可以获得涉及事务并发性的脏读数据，如下面进一步讨论的。如本文中所使用的“脏读”在事务读取尚未提交的数据时发生。例如，假设

事务1更新数据库104的一个行。事务2在事务1提交该更新之前读取更新的行。这种情况被认为对应于“脏读”。

[0127] 在一个实施例中，记录引擎202不记录由删除操作(“DLT”)产生的数据的数据映像。

[0128] 在一个实施例中，记录引擎202将SQL数据定义语言(DDL)语句记录为由DDL操作产生的数据映像。在一个实施例中，“DDL”操作用于定义数据结构，诸如结构化查询语言(SQL)中的创建表、更改表、删除表、创建视图等。

[0129] 在一个实施例中，记录引擎202将SQL表达式记录为由批量数据操纵语言(DML)操作产生的数据映像。在一个实施例中，DML操作用于操纵数据本身。DML操作的示例包括SQL中的插入、更新或删除指令。

[0130] 如果适用，由记录引擎202用来记录数据映像的软件工具的示例包括但不限于Equalum®、Hevo Data、HVR、IBM® WebSphere®、Qlik®、Oracle® GoldenGate®、Precisely®、Striim®、Talend®等。

[0131] 在记录引擎202记录或不记录由数据操作产生的数据的数据映像后，在操作407中，数据库管理系统102的分析引擎201基于时间序列数据，将数据库104的分析的表、列表或索引的记录或行的查询链接到分析的表、列表或索引的记录或行的先前执行的查询(如果有的话)。例如，由计算设备101的用户发出的请求从数据库104的表、列表或索引的特定记录或行访问数据(诸如操纵或检索数据)的查询可以发生在时间T3。访问数据库104的表、列表或索引的相同记录或行的在先查询可以发生在时间T2，该时间T2可以在查询在时间T1访问数据库104的表、列表或索引的相同记录或行之后发生。在时间T3发生的查询然后可以链接到在时间T2和T1发生的查询。

[0132] 如上所述，在一个实施例中，这样的链接可以经由由分析引擎201提供的指针来完成，所述指针诸如与在时间T3的查询相关联的结构化审计日志中的指针，所述指针指向与在时间T2的查询相关联的结构化审计日志，该结构化审计日志包含指向与在时间T1的查询相关联的结构化审计日志的指针。以这种方式，可以容易地检索历史查询的结果集(如果适用，具有元数据和数据映像的结构化审计日志)。

[0133] 在一个实施例中，分析引擎201基于与查询相关联的记录或行标识符(“RID”)确定这种链接。例如，在时间T1、T2和T3发生的查询可以全部与相同的RID(诸如12345)相关联。在一个实施例中，分析引擎201例如在数据结构(例如，表)中存储数据库104的所分析的表、列表或索引的RID。在一个实施例中，分析引擎201还将与RID相关联的时间序列数据存储在数据结构中。这样的时间序列数据可以用于链接与过去发生的与相同RID相关联的查询相关联的结构化审计日志(包含如下文进一步讨论的时间序列数据)。在一个实施例中，这样的数据结构被存储在数据库管理系统102的存储设备(例如，存储器305、盘单元308)中。

[0134] 在操作408中，数据库管理系统102的审计日志创建器203创建结构化审计日志以及将审计信息存储在所创建的结构化审计日志内，诸如由分析引擎201提取和识别的元数据、由记录引擎202记录的原始数据映像(如果有的话)、以及由分析引擎201提供的指向与历史查询相关联的结构化审计日志的任何指针(即，指向对与所创建的结构化审计日志相关联的数据库102的所分析的表、列表或索引的相同记录或行的先前执行的查询的指针)。在一个实施例中，这样的结构化审计日志以及存储在结构化审计日志中的审计信息与所分

析的表、列表或索引的记录或行的记录或行标识符相关联。

[0135] 如以上所讨论的，如在此使用的“结构化审计日志”是指存储由分析引擎201提取和识别的元数据、由记录引擎202记录的原始数据映像(如果有的话)、以及由分析引擎201提供的指向与历史查询相关联的结构化审计日志的任何指针的数据结构。在一个实施例中，此类结构化审计日志被存储在数据库管理系统102的存储设备(例如，存储器305、盘单元308)中。

[0136] 由审计日志创建器203用来创建结构化审计日志并且将以上讨论的审计信息存储在这种创建的结构化审计日志中的软件工具的示例包括但不限于erwin® Data Modeler、ER/Studio®、DbSchema、ERBuilder、HeidiSQL、Navicat® Data Modeler、Toad Data Modeler、Archi等。

[0137] 以下结合图5-8讨论在结构化审计日志中存储元数据和记录的数据映像(如果有的话)连同指向任何先前执行的查询的指针的图示。

[0138] 如以上所讨论的，在一个实施例中，记录引擎202记录在执行所述操作之后从更新(“UPD”)或插入(“INS”)操作产生的数据的数据映像，如在图5中所示出的。

[0139] 图5示出了根据本公开的实施例的涉及更新操作的结构化审计日志501。

[0140] 如以下将进一步详细讨论的，所提取的元数据(包括任何记录的数据映像)将被存储在结构化审计日志(诸如图5的结构化审计日志500)中。

[0141] 如图5所示，结构化审计日志501与记录或行标识符(RID)502相关联。在一个实施例中，结构化审计日志501包括由分析引擎201针对由RID 502标识的数据库104的所分析的表、列表或索引的记录或行识别的元数据字段503。例如，元数据503包括时间序列(TS)数据504、用户数据505、IP地址506(诸如数据库104的IP地址506)以及数据操作(“OP”)507。如图5所示，TS数据504对应于TS3的时间。用户数据505对应于用户#1(“USR1”)。数据库104的IP地址506对应于“IP1”(例如，192.158.1.38)。数据操作507对应于更新操作(“UPD”)。

[0142] 此外，在一个实施例中，结构化审计日志501包括原始数据映像字段508，其可被用于存储记录的数据映像。

[0143] 在一个实施例中，为了节省存储空间，记录引擎202如图5所示记录在执行所述操作之后由更新(“UPD”)操作产生的数据的数据映像508。例如，这样的数据映像508作为“Img1v3”被记录并保存在结构化审计日志501中。

[0144] 此外，如图5所示，结构化审计日志501包括指向结构化审计日志510的指针509，结构化审计日志510与结构化审计日志501类似地结构化。如上所述，时间序列数据可以用于链接涉及相同RID的结构化审计日志。例如，结构化审计日志501链接到结构化审计日志510，因为结构化审计日志501与结构化审计日志510两者均与相同的RID相关联，并且结构化审计日志510与刚好在时间T3(参见结构化审计日志501的时间序列数据字段504)之前出现的时间T2(参见结构化审计日志510的时间序列数据字段504)相关联。类似地，如图5所示，结构化审计日志510包括指向结构化审计日志512(与结构化审计日志501类似地结构化)的指针511，结构化审计日志512涉及与结构化审计日志501相同的RID，但是与刚好在时间T2(参见结构化审计日志510的时间序列数据字段504)之前出现的时间T1(参见结构化审计日志512的时间序列数据字段504)相关联。

[0145] 此外，如图5中所示，记录引擎202记录在执行所述操作之后由插入(“INS”)操作得

到的数据的数据映像(参见结构化审计日志512的数据操作字段507)。例如,这样的数据映像被记录并保存在结构化审计日志512中的原始数据映像字段508中作为“Img1v1”。

[0146] 如以上所讨论的,为了节省存储空间,记录引擎202记录在执行所述操作之后由更新(“UPD”)操作产生的数据的数据映像。为了在更新操作之前获得数据的映像,可如上所述经由指针利用至先前映像的链接。

[0147] 在一个实施例中,此类记录的数据映像可仅包括数据库104的表、列表或索引的改变的行或列的数据。

[0148] 在一个实施例中,记录引擎202除了当存在并发事务时不记录由读取/查询操作(“QRY”)产生的数据的数据映像,如图6所示。以这种方式,事务并发中的脏读数据可以如下文进一步讨论的那样获得。如本文中所使用的“脏读”在事务读取尚未提交的数据时发生。例如,假设事务1更新数据库104的一行。事务2在事务1提交更新之前读取更新的行。这种情况被认为对应于“脏读”。

[0149] 参见图6,图6示出了根据本公开的实施例的涉及读取或查询操作的结构化审计日志。

[0150] 如先前结合图5所讨论的,结构化审计日志501与记录或行标识符(RID)502相关联。这样的结构化审计日志501可以与指向结构化审计日志602(与图5的结构化审计日志501类似地结构化)的指针601相关联,如图6中所示。结构化审计日志501链接到结构化审计日志602,因为两者与相同的RID相关联,并且结构化审计日志602与刚好在时间T3(参见结构化审计日志501的时间序列数据字段504)之前出现的时间T2(参见结构化审计日志602的时间序列数据字段504)相关联。

[0151] 在一个实施例中,记录引擎202通常不记录读取/查询操作(“QRY”)的原始数据映像。然而,当发生并发事务时,诸如图6中所示,记录引擎202记录读取/查询操作的得到的映像,其由审计日志创建器203存储在结构化审计日志中。

[0152] 例如,如图6所示,结构化审计日志602、603和604(全部与结构化审计日志501类似地结构化)全部与时间序列数据字段504中的时间序列数据TS2相关联。如结构化审计日志603中所示,发生数据库104的更新操作,这导致在执行如以上结合图5所讨论的更新(“UPD”)操作之后记录所述操作产生的数据的数据映像(参见原始数据映像字段508的“Img1v2”)。然而,如结构化审计日志604中所示,可能在提交更新操作之前执行如数据操作507字段中所示的读取操作(“QRY”)。因此,结构化审计日志602存储读取操作的记录的数据映像(参见原始数据映像字段508的“Img1v1”)。以这种方式,可以获得涉及事务并发的脏读数据。

[0153] 此外,如图6所示,结构化审计日志602具有指向结构化审计日志603的指针605,结构化审计日志603具有指向结构化审计日志604的指针606。这些结构化审计日志602、603、604中的每一个发生在时间T2。具体地,结构化审计日志603、604是并发事务,因为两者同时发生。如以上所讨论的,当发生事务并发时,诸如在提交更新操作之前执行的读取/查询操作,创建结构化审计日志(诸如结构化审计日志602)来存储读取操作的原始数据映像,如以上所解释的。

[0154] 此外,如图6所示,结构化审计日志604包括指针607以链接结构化审计日志608(与图5的结构化审计日志501类似地结构化)。如上所述,时间序列数据可以用于链接涉及相同

RID的结构化审计日志。例如,结构化审计日志608链接到结构化审计日志604,因为两者与相同的RID相关联,并且结构化审计日志608与刚好在时间T2(参见结构化审计日志604的时间序列数据字段504)之前出现的时间T1(参见结构化审计日志608的时间序列数据字段504)相关联。

[0155] 此外,在一个实施例中,记录引擎202不记录由删除操作("DLT")产生的数据的数据映像。

[0156] 在一个实施例中,记录引擎202将SQL数据定义语言(DDL)语句记录为由DDL操作产生的数据映像,如图7所示。在一个实施例中,"DDL"操作用于定义数据结构,诸如结构化查询语言(SQL)中的创建表、删除表、更改表、删除表、创建视图等。

[0157] 图7示出了根据本公开的实施例的涉及数据定义语言(DDL)操作的结构化审计日志701。

[0158] 参见图7,结构化审计日志701(与图5的结构化审计日志501类似地结构化)由审计日志创建器203创建以存储通过分析引擎201获得的元数据以及由记录引擎202记录的数据映像。如图7所示,数据操作字段507的数据操作对应于"alter table"(更改表),这是DDL操作。在这种情况下,记录引擎202记录如原始数据映像字段508中所示的SQL DDL语句。

[0159] 此外,如图7所示,结构化审计日志701经由指针703链接到结构化审计日志702。如上所述,时间序列数据可以用于链接涉及相同RID的结构化审计日志。例如,结构化审计日志701链接到结构化审计日志702,因为结构化审计日志701和结构化审计日志702都与相同的RID相关联,并且结构化审计日志702与刚好在时间T3(参见结构化审计日志701的时间序列数据字段504)之前出现的时间T2(参见结构化审计日志702的时间序列数据字段504)相关联。

[0160] 如图7所示,结构化审计日志702还包括数据操作字段507中的DDL操作("create view"),这导致记录引擎202记录如结构化审计日志702的原始数据映像字段508中所示的SSL DDL语句。

[0161] 在一个实施例中,记录引擎202将SQL表达式记录为由批量数据操纵语言(DML)操作产生的数据映像,如图8所示。在一个实施例中,DML操作用于操纵数据本身。DML操作的示例包括SQL中的插入、更新或删除指令。

[0162] 图8示出了根据本公开的实施例的涉及批量数据操纵语言(DML)操作的结构化审计日志。

[0163] 参见图8,在一个实施例中,一次用该表达式跟踪批量操作,诸如批量DML操作。例如,批量DML操作可包括更新多个记录。

[0164] 如图8所示,结构化审计日志801(与图5的结构化审计日志501类似地结构化)与两个RID 502(即RID＝1 802以及RID＝2 803)相关联,其表示数据库104的表、列表或索引的多个记录或行,它们涉及批量DML操作(多个更新操作)而被更新。在一个实施例中,审计日志创建器203将与批量DML操作相关联的SQL DDL语句("Col1*0.21+Col2")存储在结构化审计日志801的原始数据映像字段508中。

[0165] 此外,如图8所示,在这样的批量DML操作中,在结构化审计日志804、805(类似于图5的结构化审计日志501被结构化)中表示多个更新操作,其中,如图8所示,经由指针807将结构化审计日志805链接到结构化审计日志806(结构类似于图5的结构化审计日志501)。如

上所述,时间序列数据可以用于链接涉及相同RID的结构化审计日志。例如,结构化审计日志805被链接到结构化审计日志806,因为两者与相同的RID相关联,并且结构化审计日志806与刚好在时间T2(参见结构化审计日志805的时间序列数据字段504)之前出现的时间T1(参见结构化审计日志806的时间序列数据字段504)相关联。

[0166] 在创建结构化审计日志后,结构化审计日志可以用于审计数据库104,如以下结合图9所讨论的。

[0167] 图9是根据本公开的实施例的用于通过利用结构化审计日志来提高数据库审计的准确性和效率的方法900的流程图。

[0168] 结合图1-8参见图9,在操作901中,数据库管理系统102的查询引擎204(例如,Presto®、Apache®Drill、Cloudera®Impala、Apache®Spark等)从计算设备101的用户接收用于审计数据库104的查询请求。

[0169] 在操作902中,数据库管理系统102的查询引擎204识别与查询请求相关联的记录或行标识符(RID)。

[0170] 如上所述,在一个实施例中,用于审计数据库104的查询包括对其执行审计的数据库104的表、列表或索引的记录或行的记录或行标识符(RID)。在一个实施例中,查询引擎204搜索与结构化审计日志相关联的RID以确定是否存在匹配。在一个实施例中,每个结构化审计日志与一RID相关联,其中这样的信息被存储在数据结构中。在一个实施例中,查询引擎204在这样的数据结构中搜索以识别任何匹配的RID。在一个实施例中,查询引擎204利用自然语言处理来识别数据结构中的任何匹配。在一个实施例中,这样的数据结构被存储在数据库管理系统102的存储设备(例如,存储器305、磁盘单元308)中。

[0171] 在一个实施例中,查询引擎204分析来自计算设备101的用户的用于审计数据库104的查询请求以确定数据库104的表、列表或索引的哪些记录或行要进行审计。例如,查询可以包括记录或行号函数或选择子句,其识别要进行审计的数据库104的表、列表或索引的记录或行。在识别数据库104的表、列表或索引的记录或行时,可以经由ROWID语句来识别记录或行标识符(RID)。

[0172] 在操作903中,由数据库管理系统102的查询引擎204确定在与查询相关联的记录或行标识符(RID)和与结构化审计日志(诸如结构化审计日志501)相关联的记录或行标识符之间是否存在匹配。

[0173] 如上所述,结构化审计日志的RID可以存储在数据结构中,例如存储在数据库管理系统102的存储设备(例如,存储器305、磁盘单元308)中。在一个实施例中,查询引擎204利用自然语言处理来识别这种数据结构中的任何匹配。

[0174] 如果在与查询相关联的RID和与结构化审计日志相关联的RID之间存在匹配,则在操作904中,数据库管理系统102的查询引擎204从如图10所示的匹配的结构化审计日志检索经审计的信息。这样的经审计的信息可以包括来自元数据字段503的信息以及来自原始数据映像字段508的任何数据映像。此外,这样的经审计的信息可以包括来自任何链接的结构化审计日志的任何经审计的信息,如下面结合图10进一步讨论的。

[0175] 在操作905中,数据库管理系统102的查询引擎204诸如经由如下文结合图10所讨论的计算设备101的用户界面将这样的经审计的信息提供给计算设备101的用户。

[0176] 图10示出了根据本公开的实施例的将与结构化审计日志相关联的记录或行标识

符(RID)和与用于审计数据库的查询相关联的RID进行匹配和不匹配。

[0177] 参见图10,查询引擎204可以在存储结构化审计日志(诸如结构化查询日志1001)的RID的数据结构中执行对与查询相关联的RID的搜索。如果查询与RID＝1相关联,并且结构化查询日志1001与RID＝1相关联(参见图10的元素1002),如在数据结构中指示的并且在图10中示出的,则查询引擎204将识别结构化查询日志1001的RID为与查询相关联的RID匹配。在将与查询相关联的RID和与结构化审计日志相关联的RID进行匹配之后,查询引擎204获得经审计的信息,诸如元数据字段503中的信息以及原始数据映像字段508中的数据映像(如果有的话)。此外,这样的经审计的信息可以包括任何链接的历史查询,诸如经由指针1004链接到结构化审计日志1001的结构化审计日志1003中的经审计的信息。此外,这样的审计信息可包括结构化审计日志1005中的审计信息,所述结构化审计日志1005经由指针1006链接到结构化审计日志1003,如图10所示。这样的匹配的结构化审计日志被认为在“RID匹配区域”1007内;然而,与不匹配与用于审计数据库104的查询相关联的RID的RID相关联的那些结构化审计日志被称为在“RID不匹配区域”1008内。

[0178] 例如,结构化审计日志1009可以不与匹配于查询引擎204接收到审计数据库104的任何查询的RID的RID相关联。如图10所示,结构化审计日志1009经由指针1011链接到结构化审计日志1010。

[0179] 返回图9的操作903,如果在与用于审计数据库104的查询相关联的RID和与结构化审计日志的RID之间不存在匹配,则在操作906中,数据库管理系统102的查询引擎204诸如经由计算设备101的用户界面通知请求者(计算设备101的用户)数据库104的表、列表或索引的记录或行没有被访问。

[0180] 作为前述内的结果,本公开的实施例提供了一种用于构建和使用结构化审计日志来准确证明数据库(例如,NoSQL数据库)的数据访问的手段。

[0181] 此外,本公开的原理改进了涉及数据库审计的技术或技术领域。如上所述,数据库审计是对所选用户数据库动作的监视和记录。它可以基于单独的动作,诸如所执行的SQL语句的类型,或诸如用户名、应用、时间等因素的组合。基于数据库审计的使用,识别数据库的哪些行和列以及由谁和何时被访问是重要的。不幸的是,这样的信息不容易被跟踪和记录,尤其是对于NoSQL(“非结构化查询语言(SQL)”)数据库。NoSQL数据库提供用于数据的存储和检索的机制,该机制以不同于在关系数据库中使用的表关系的方式被建模。NoSQL数据库使用的数据结构(例如,键-值对)与关系数据库中默认使用的数据结构不同,从而使得一些操作在NoSQL中更快。此外,NoSQL数据库利用非结构化存储,其允许大规模地高性能、灵活的信息处理。例如,NoSQL数据库可以跨多个处理节点以及跨多个服务器存储非结构化数据。然而,通过跨多个处理节点以及跨多个服务器来存储非结构化数据,这使得对这样的数据库的审计成为挑战。当前,可以利用审计插件来尝试监视和记录数据库动作,诸如用于NoSQL数据库。不幸的是,仅数据操作(例如,更新操作、读取操作)被记录。作为监视和记录有限数据的结果,审计在识别数据库的哪些行和列被访问以及由谁和何时被访问方面是有缺陷的。可替代地,时间表可以用于尝试监视和记录数据库动作,诸如用于NoSQL数据库。不幸的是,一些操作未被记录。如在使用审计插件的场景中,时间表的使用导致有限数据被监视和记录,从而导致审计在识别数据库的哪些行和列以及由谁和何时被访问方面是有缺陷的。此外,时间表的使用是耗时且低效的,尤其涉及对数据库的频繁读取、添加、删除和更新

操作。另外，在事务并发性(两个事务一起运行，诸如在重叠的时间段期间访问相同的数据库行)中获得特定信息(诸如脏读数据)的请求可能无法使用时间表来实现。结果，当前不存在用于以高效方式有效地审计数据库动作的手段，其能够实现对数据库(例如，NoSQL数据库)的数据访问的准确证明。

[0182] 本公开的实施例通过分析数据库的表、列表或索引以识别元数据来改进这种技术，所述元数据包括时间序列数据、用户数据、互联网协议地址和操作数据。如本文中所使用的“元数据”是指提供关于其他数据的信息的数据。如在本文中所使用的“时间序列数据”是指发出查询以对来自数据库的表、列表或索引的信息进行更新、插入、删除等的特定顺序。如本文所使用的“用户数据”是指发出查询以访问数据库的表、列表或索引的记录或行的特定用户的标识符。如在此所使用的“互联网协议(IP)地址”是指分配给设备(诸如数据库)的唯一标识符。如本文所使用的“操作数据”是指通过由计算设备的用户发出的查询请求在数据库上执行的操作。在一个实施例中，这样的操作包括更新操作、删除操作、读取操作、数据定义语言操作(用于定义数据结构，例如结构化查询语言中的创建表、更改表、删除表、创建视图等)、批量数据操纵语言操作(用于操纵数据库的表、列表或索引的多个记录或行)等。所识别的元数据与从中提取元数据的数据库的表、列表或索引的对应记录或行相关联。然后，基于对应的数据操作，确定是否记录与分析的数据库的表、列表或索引的记录或行相关联的原始数据映像。如本文中所使用的“原始数据映像”或“数据映像”是指由数据操作(例如，更新操作)产生的数据的映像。例如，可以响应于更新或插入操作来记录从数据操作得到的数据的数据映像。然而，除存在并发事务时之外，可以不响应于删除操作或响应于读取/查询操作来记录从数据操作得到的数据的数据映像。然后，将所识别的元数据以及所记录的数据映像(如果有的话)存储在“结构化审计日志”中，该“结构化审计日志”与在其上获得这种结构化信息的数据库的分析的表、列表或索引的记录或行的记录或行标识符相关联。如本文所使用的“结构化审计日志”是指存储所述审计信息(例如，元数据、记录的数据映像等)的数据结构。在一个实施例中，结构化审计日志被存储在数据库管理系统的存储设备(例如，存储器、磁盘单元)中，其中用于审计数据库的未来查询可以访问这样的存储的结构化审计日志以获得关于数据库的审计信息。在一个实施例中，基于将与查询相关联的行或记录标识符和与包含所请求的数据库审计信息的结构化审计日志相关联的记录或行标识符匹配，从所存储的结构化审计日志获得适当的审计信息。以这种方式，通过利用结构化审计日志来提高数据库审计的准确性和效率。此外，以这种方式，在涉及数据库审计的技术领域中存在改进。

[0183] 本公开提供的技术方案不能在人的头脑中执行，或由人使用笔和纸来执行。即，在不使用计算机的情况下，在人的脑海中或者通过人使用笔和纸在任何合理的时间量内并且在任何合理的准确性预期下都不能实现本公开提供的技术方案。

[0184] 在本公开的一个实施例中，一种用于提高数据库审计的准确性和效率的计算机实现的方法包括分析数据库的表、列表或索引以识别元数据，其中所述元数据包括时间序列数据、用户数据、互联网协议(IP)地址和操作数据。该方法还包括将所识别的元数据与所分析的数据库的表、列表或索引的对应记录或行相关联。该方法另外包括基于对应的数据操作来确定是否记录与该数据库的所分析的表、列表或索引的记录或行相关联的数据映像。此外，该方法包括将所识别的元数据和所记录的数据映像(如果有的话)存储在结构化审计

日志中，所述结构化审计日志与该数据库的所分析的表、列表或索引的记录或行的记录或行标识符相关联。

[0185] 此外，在本公开的一个实施例中，所述方法进一步包括当存在并发事务时，记录由读取操作产生的数据的映像。另外，该方法包括当存在并发事务时响应于将由读取操作产生的数据的记录映像存储在结构化审计日志中，从结构化审计日志获得涉及事务并发性的脏读数据。

[0186] 此外，在本公开的一个实施例中，该方法进一步包括将结构化查询语言数据定义语言语句记录为数据定义语言操作的数据映像。该方法另外包括将结构化查询语言表达式记录为批量数据操纵语言操作的数据映像。

[0187] 此外，在本公开的一个实施例中，该方法进一步包括将该数据库的分析的表、列表或索引的记录或行的查询链接到该数据库的分析的表、列表或索引的该记录或行的先前执行的查询。

[0188] 进一步地，在本公开的一个实施例中，该方法进一步包括在所述结构化审计日志中存储指针，以将所述数据库的分析的表、列表或索引的记录或行的查询链接到所述数据库的分析的表、列表或索引的所述记录或行的先前执行的查询。

[0189] 此外，在本公开的一个实施例中，该方法进一步包括接收对数据库进行审计的查询请求。所述方法还包括结合对数据库进行审计的查询请求来识别数据库的表、列表或索引的记录或行标识符。

[0190] 此外，在本公开的一个实施例中，该方法进一步包括响应于所识别的与查询请求相关联的记录或行标识符和与结构化审计日志相关联的记录或行标识符之间的匹配，从结构化审计日志检索经审计的信息。

[0191] 上述方法的实施例的其他形式是系统和计算机程序产品。

[0192] 已经出于说明的目的呈现了本公开的不同实施例的描述，但并不旨在是详尽的或限于所公开的实施例。在不脱离所描述的实施例的范围和精神的情况下，许多修改和变化对本领域普通技术人员将是显而易见的。这里使用的术语被选择来最好地解释实施例的原理、实际应用或对在市场中找到的技术的技术改进，或者使得本领域普通技术人员能够理解这里公开的实施例。

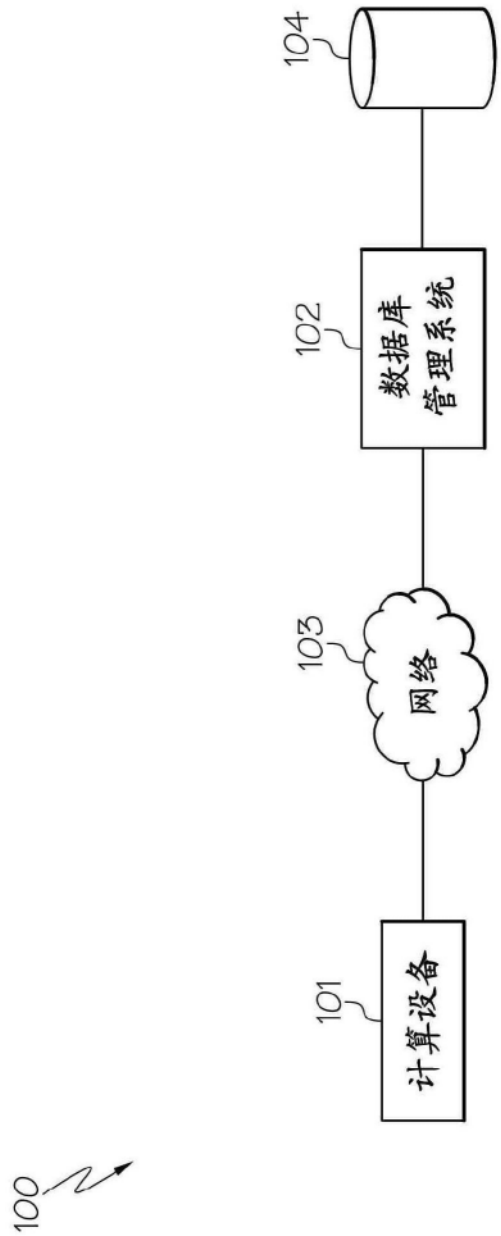

104
102
数据库管理系统
103
网络
101
计算设备
100

图1

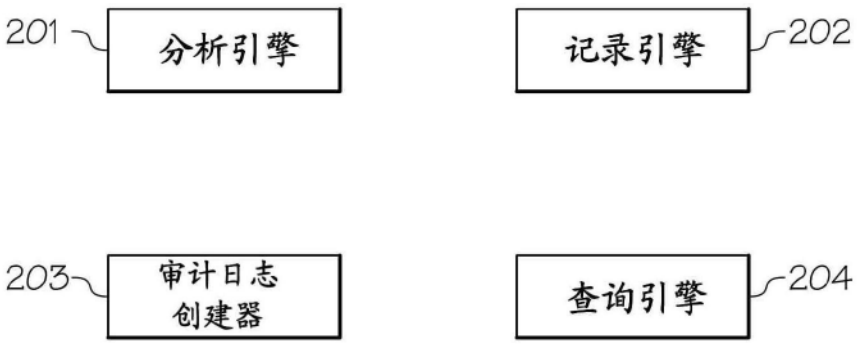

201
分析引擎
记录引擎
202
203
审计日志创建器
查询引擎
204

图2

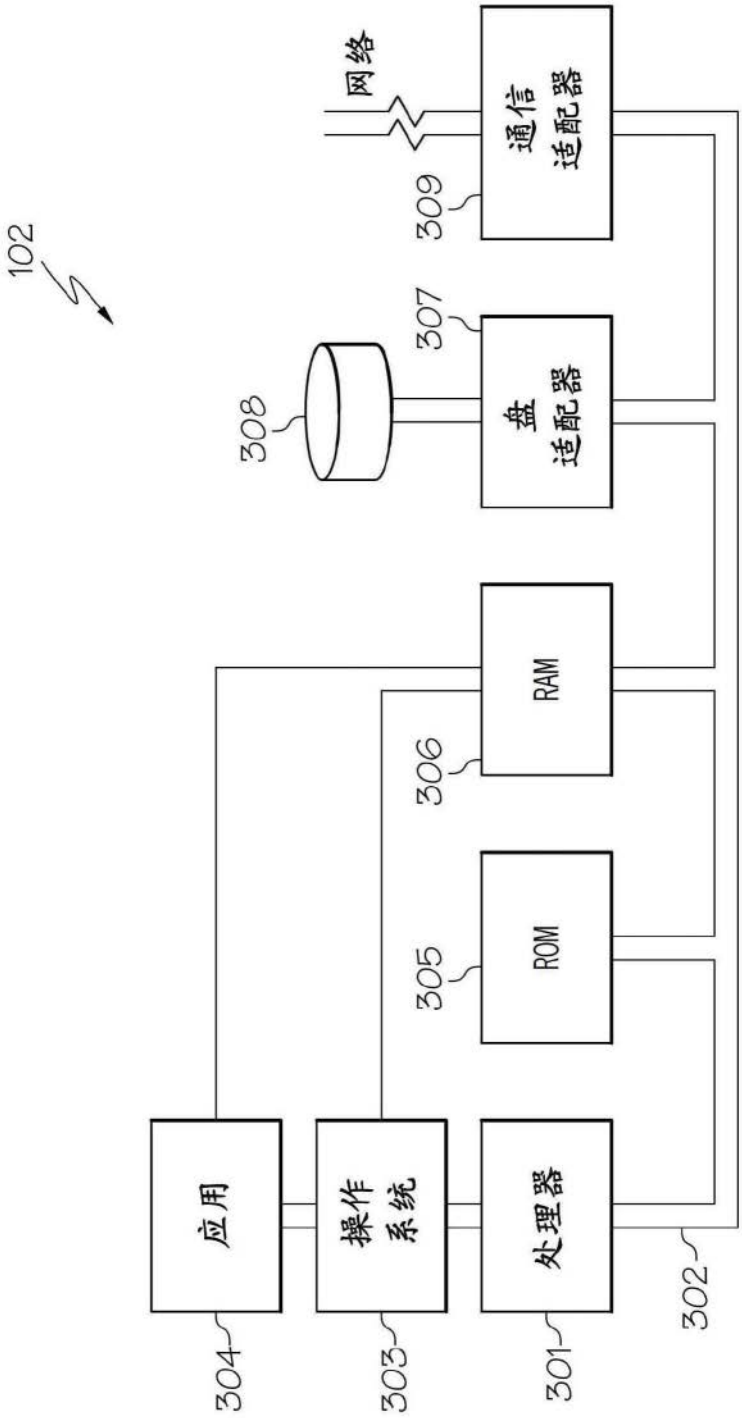

网络
通信适配器
309
102
307
盘适配器
308
RAM
306
ROM
305
应用
操作系统
处理器
304
303
301
302

图3

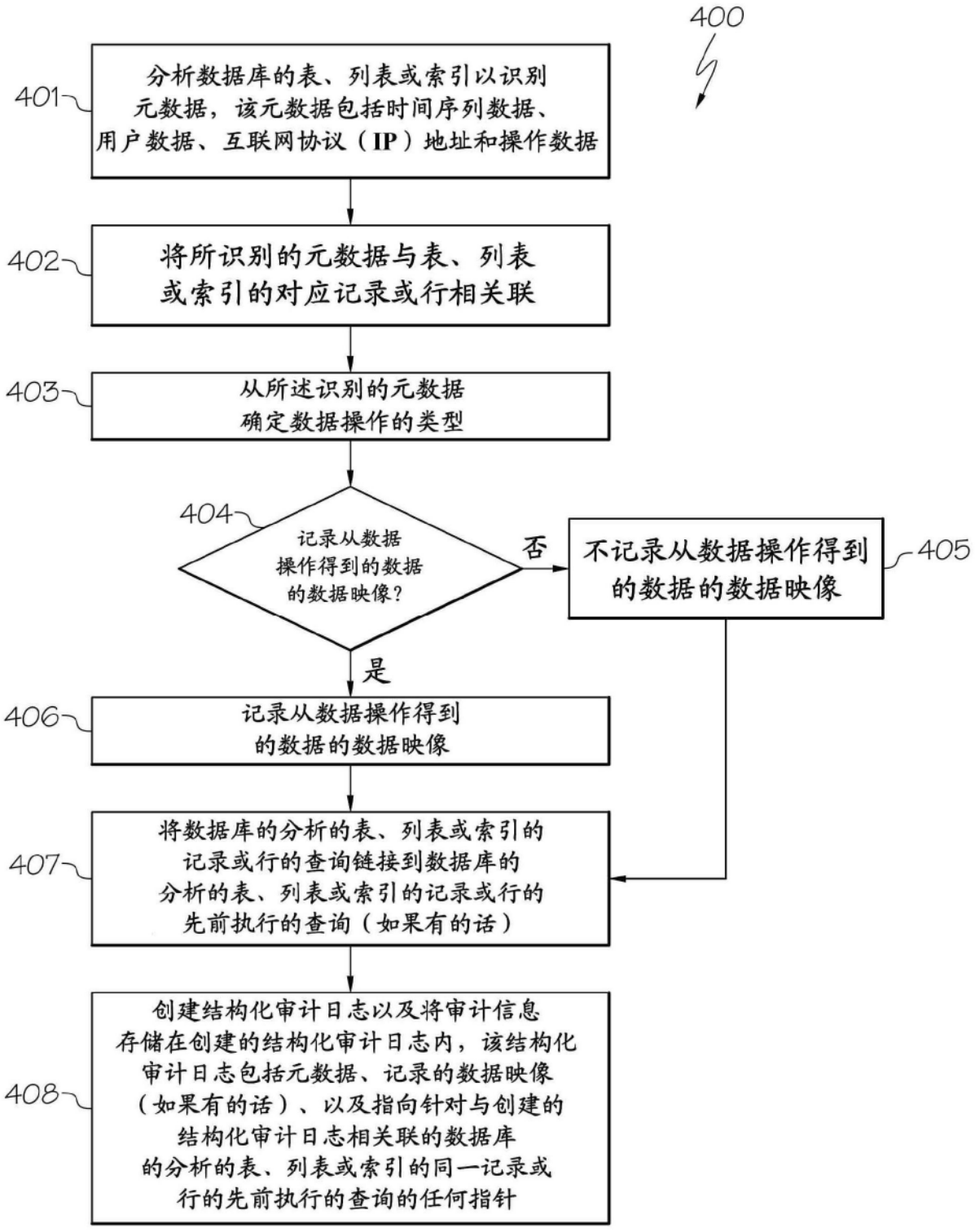

400
401
分析数据库的表、列表或索引以识别元数据，该元数据包括时间序列数据、用户数据、互联网协议（IP）地址和操作数据
402
将所识别的元数据与表、列表或索引的对应记录或行相关联
403
从所述识别的元数据确定数据操作的类型
404
记录从数据操作得到的数据的数据映像？
否
405
不记录从数据操作得到的数据的数据映像
是
406
记录从数据操作得到的数据的数据映像
407
将数据库的分析的表、列表或索引的记录或行的查询链接到数据库的分析的表、列表或索引的记录或行的先前执行的查询（如果有的话）
408
创建结构化审计日志以及将审计信息存储在创建的结构化审计日志内，该结构化审计日志包括元数据、记录的数据映像（如果有的话）、以及指向针对与创建的结构化审计日志相关联的数据库的分析的表、列表或索引的同一记录或行的先前执行的查询的任何指针

图4

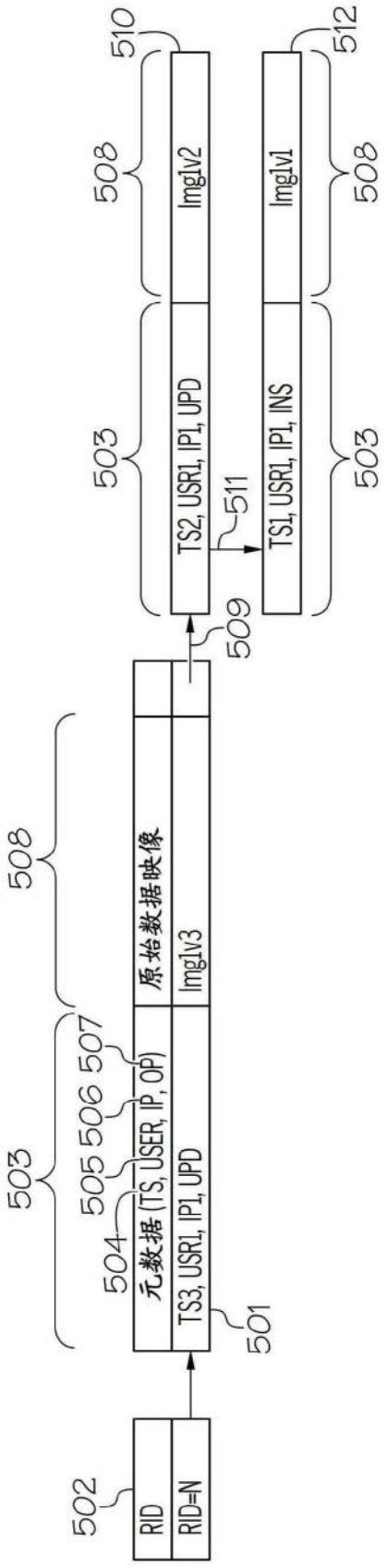

502
RID
RID=N
501
TS3, USR1, IP1, UPD
504
元数据(TS, USER, IP, OP)
505
506
507
原始数据映像
Img1v3
503
508
509
511
TS2, USR1, IP1, UPD
TS1, USR1, IP1, INS
Img1v2
Img1v1
510
512

图5

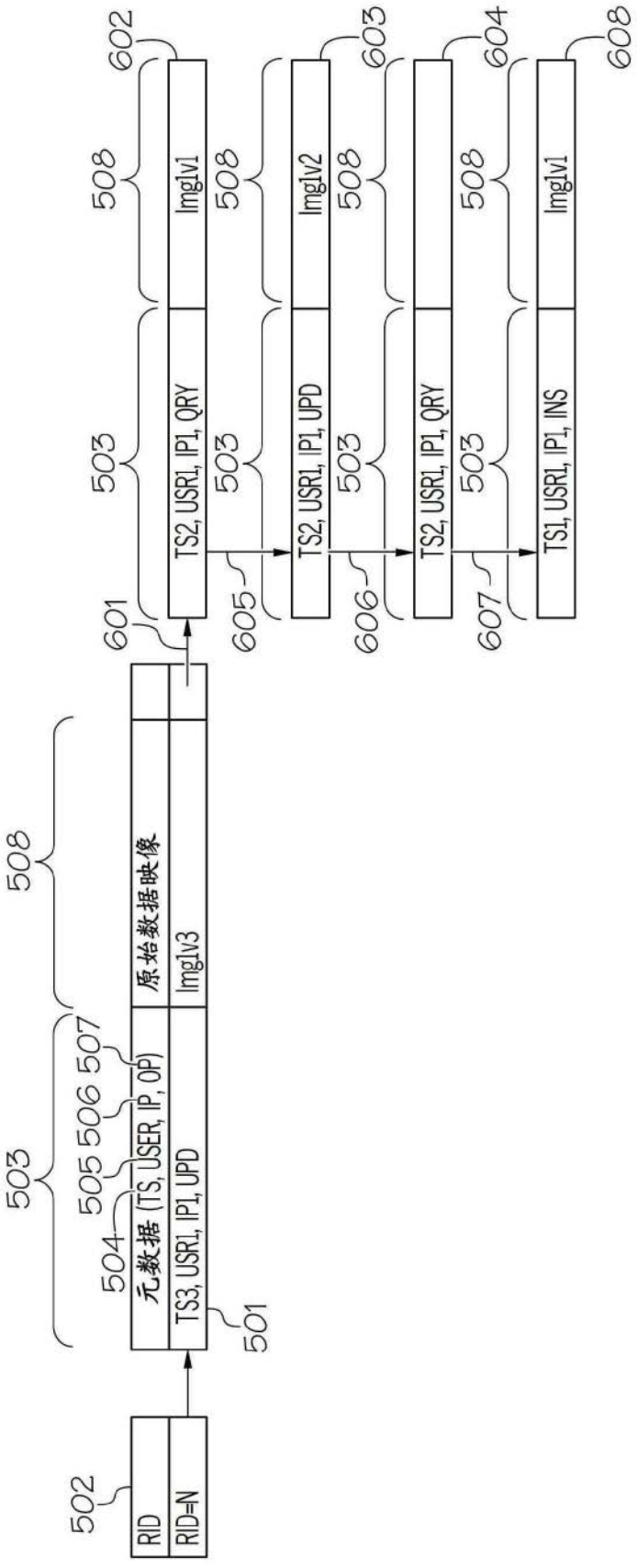

RID
RID=N
TS3, USR1, IP1, UPD
元数据 (TS, USER, IP, OP)
原始数据映像
Img1v3
TS2, USR1, IP1, QRY
Img1v1
TS2, USR1, IP1, UPD
Img1v2
TS2, USR1, IP1, QRY
TS1, USR1, IP1, INS
Img1v1
501
502
503
504
505
506
507
508
601
602
603
604
605
606
607
608

图6

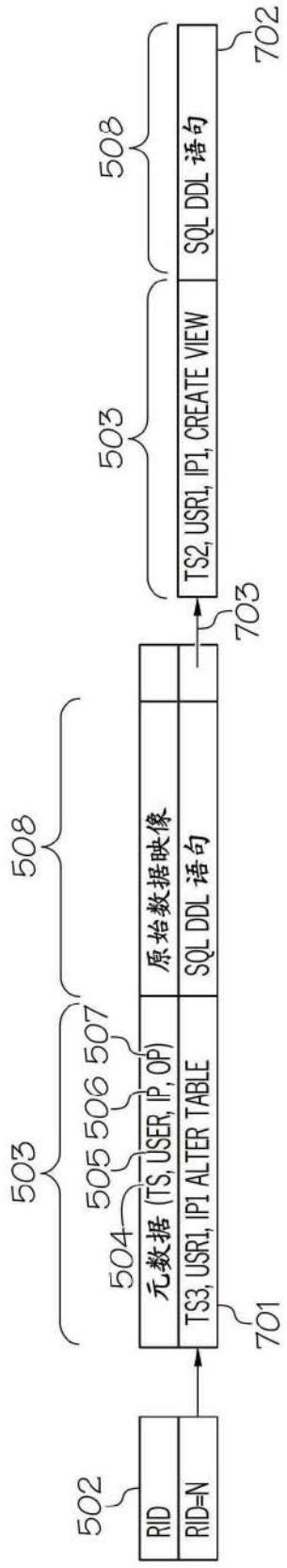

502
RID
RID=N
504
元数据 (TS, USER, IP, OP)
505
506
507
503
508
TS3, USR1, IP1 ALTER TABLE
701
原始数据映像
SQL DDL 语句
703
503
508
TS2, USR1, IP1, CREATE VIEW
SQL DDL 语句
702

图7

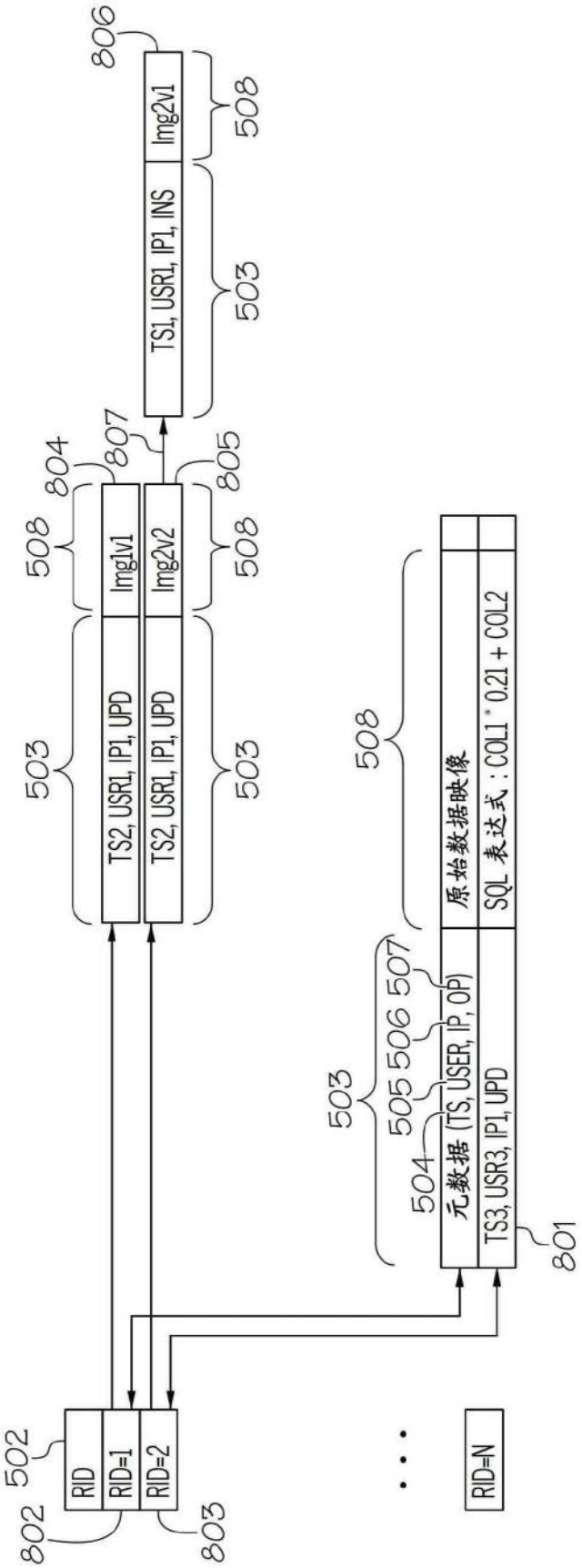

806
508
Img2v1
TS1, USR1, IP1, INS
503
804
807
805
508
Img1v1
Img2v2
508
TS2, USR1, IP1, UPD
TS2, USR1, IP1, UPD
503
503
508
原始数据映像
SQL 表达式：COL1 * 0.21 + COL2
507
506
505
503
504
元数据 (TS, USER, IP, OP)
TS3, USR3, IP1, UPD
801
502
RID
RID=1
RID=2
RID=N
802
803

图8

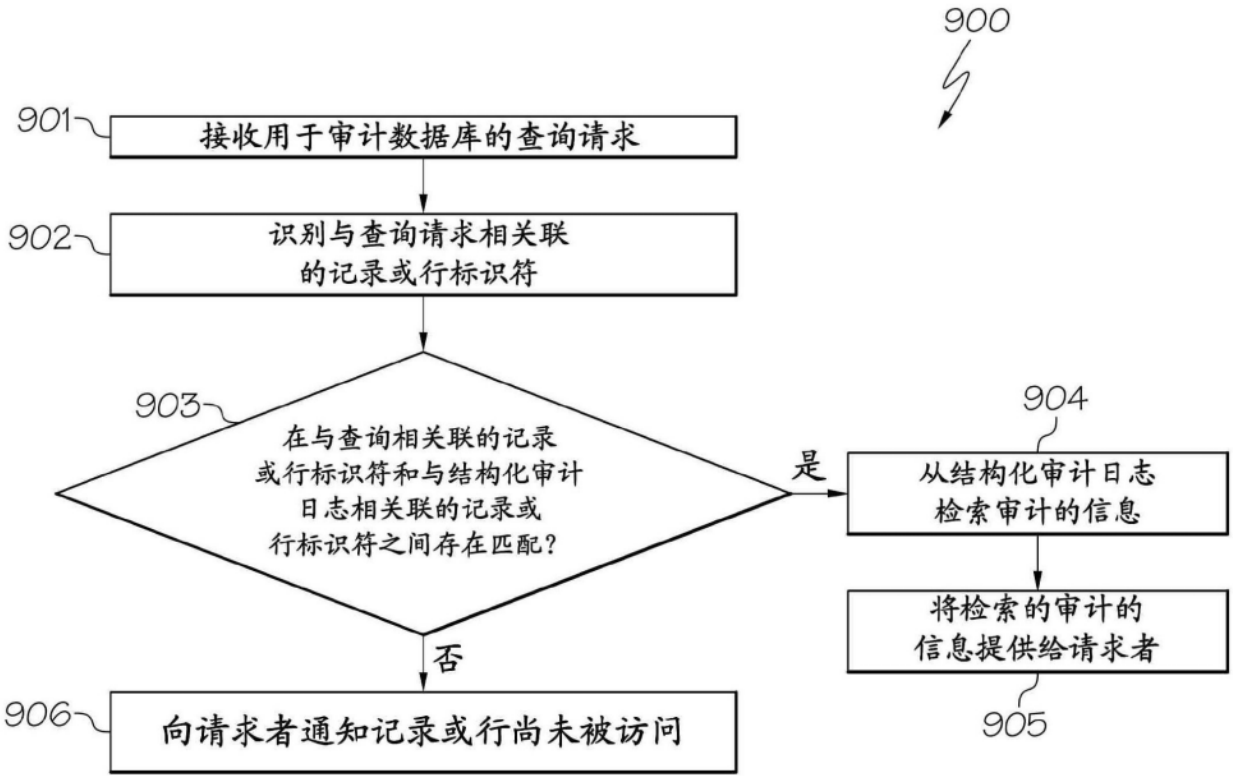

900
901
接收用于审计数据库的查询请求
902
识别与查询请求相关联
的记录或行标识符
903
在与查询相关联的记录
或行标识符和与结构化审计
日志相关联的记录或
行标识符之间存在匹配?
是
否
904
从结构化审计日志
检索审计的信息
905
将检索的审计的
信息提供给请求者
906
向请求者通知记录或行尚未被访问

图9

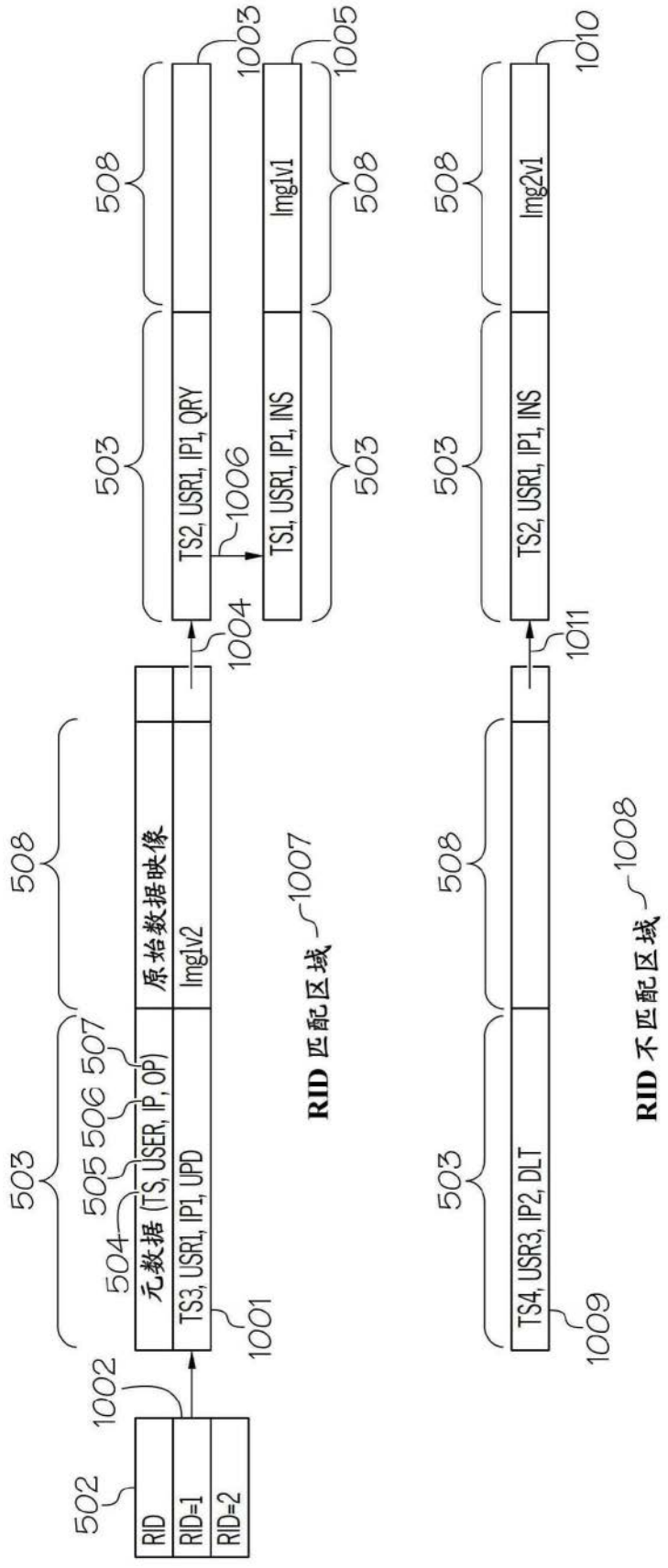

RID
RID=1
RID=2
502
1002
元数据 (TS, USER, IP, OP)
TS3, USR1, IP1, UPD
原始数据映像
Img1v2
504
505
506
507
503
508
1001
RID 匹配区域
1007
TS2, USR1, IP1, QRY
TS1, USR1, IP1, INS
Img1v1
1003
1004
1005
1006
TS4, USR3, IP2, DLT
TS2, USR1, IP1, INS
Img2v1
1009
1010
1011
RID 不匹配区域
1008

图10