(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111971666 B
(45)授权公告日 2024.10.29

(21)申请号 201980008847.5
(22)申请日 2019.01.16
(65)同一申请的已公布的文献号
申请公布号 CN 111971666 A
(43)申请公布日 2020.11.20
(30)优先权数据
62/617,970 2018.01.16 US
62/747,642 2018.10.18 US
16/248,061 2019.01.15 US
(85)PCT国际申请进入国家阶段日
2020.07.16
(86)PCT国际申请的申请数据
PCT/US2019/013827 2019.01.16
(87)PCT国际申请的公布数据
WO2019/143705 EN 2019.07.25

(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 H·布塔尼
(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 李晓芳

(51)Int.Cl.
*G06F 16/2453*(2006.01)
*G06F 16/28*(2006.01)
(56)对比文件
US 2013173528 A1,2013.07.04
US 2015370865 A1,2015.12.24
US 6014656 A,2000.01.11
审查员 卜庆庆

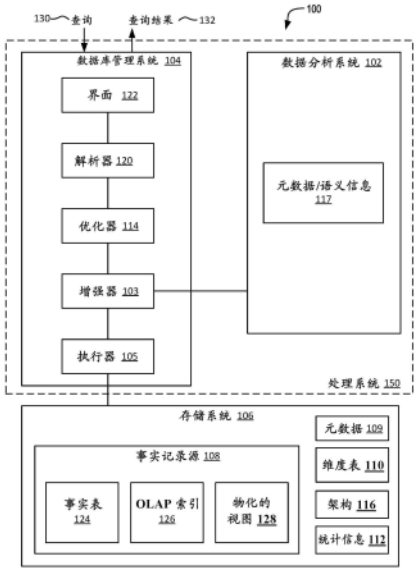
权利要求书3页 说明书41页 附图23页

(54)发明名称

# 优化SQL查询计划的维度上下文传播技术

(57)摘要

用于高效执行查询的技术。为查询生成的查询计划被优化并被重写为增强的查询计划,该增强的查询计划在被执行时使用比原始查询计划少的CPU周期并且因此比原始查询计划执行得快。因此,为其生成增强的查询计划的查询更快地执行,而不会危及获得的结果或正被查询的数据。优化包括识别原始查询计划中的一个或多个事实扫描操作的集合,然后在重写的增强的查询计划中将一个或多个维度上下文谓词条件与事实扫描操作的所述集合中的一个或多个相关联。与原始查询计划相比,这减少了在增强的查询计划中扫描和/或处理事实记录的总成本,并使增强的查询计划比原始查询计划执行得快。




CN 111971666 B

1.一种用于生成增强的查询计划的方法,包括:

由计算系统接收为用于查询关系数据库中存储的数据的查询而生成的原始查询计划,原始查询计划包括被配置为从事实记录的源扫描事实记录的第一事实扫描操作,原始查询计划还包括被配置为从第一事实扫描操作接收记录集的第二操作;

由计算系统识别要与第一事实扫描操作相关联的第一维度上下文谓词条件,其中所述第一维度上下文谓词条件对应于增强的查询计划中的第一事实扫描操作,所述增强的查询计划被配置为扫描所述事实记录的源的一个或多个记录中的事实;以及

由计算系统重写原始查询计划以生成所述增强的查询计划,其中,在增强的查询计划中,第一维度上下文谓词条件与第一事实扫描操作相关联,并且第二操作仅接收满足第一维度上下文谓词条件的那一个或多个被扫描的事实记录,

其中,由于第一维度上下文谓词条件与第一事实扫描操作的关联,增强的查询计划比原始查询计划执行得快。

2.如权利要求1所述的方法,其中用于查询的数据是非预聚合的数据。

3.如权利要求1或2所述的方法,还包括:

执行增强的查询计划以获得用于查询的结果记录集;以及

提供结果记录集作为对查询的响应。

4.如权利要求1或2所述的方法,其中,执行增强的查询计划比执行原始查询计划花费少的中央处理单元(CPU)周期。

5.如权利要求1或2所述的方法,其中,在增强的查询计划中由第二操作接收并处理的事实记录的数量小于在原始查询计划中由第二操作接收并处理的事实记录的数量。

6.如权利要求1或2所述的方法,其中所述识别要与第一事实扫描操作相关联的第一维度上下文谓词条件包括:

识别原始查询计划中的第一事实扫描操作,第一事实扫描操作对第一事实表进行操作;以及

从第一事实扫描操作开始,遍历原始查询计划以识别用于第一事实扫描操作的一个或多个适用的维度上下文谓词条件的列表,所述一个或多个适用的维度上下文谓词条件的列表包括第一维度上下文谓词条件。

7.如权利要求1或2所述的方法,其中所述识别要与第一事实扫描操作相关联的第一维度上下文谓词条件包括:

识别在原始查询计划中对第一事实表进行操作的第一事实扫描操作;

识别在原始查询计划中对第二事实表进行操作的第二事实扫描操作;

识别在原始查询计划中第二事实表和维度表之间的联接操作,其中第一维度上下文谓词条件与维度表相关联;以及

识别第一事实表和第二事实表之间的公共维度;以及

其中第一维度上下文谓词条件基于来自公共维度的属性。

8.如权利要求1或2所述的方法,其中所述识别要与第一事实扫描操作相关联的第一维度上下文谓词条件包括:

识别适用于第一事实扫描操作的多个维度上下文谓词条件;

为所述多个维度上下文谓词条件中的每个维度上下文谓词条件计算净收益度量,其中

针对所述多个维度上下文谓词条件中的维度上下文谓词条件的净收益度量是处理来自事实记录的源中的事实行的成本减去将所述维度上下文谓词条件应用于第一事实扫描操作的成本的减少的度量;

基于为所述多个维度上下文谓词条件计算的净收益度量,针对第一事实扫描操作对所述多个维度上下文谓词条件进行排序;以及

基于该排序,从所述多个维度上下文谓词条件中选择要与第一事实扫描操作相关联的维度上下文谓词条件。

9.如权利要求1或2所述的方法,其中所述识别要与第一事实扫描操作相关联的第一维度上下文谓词条件包括:

为第一事实扫描操作识别适用的维度上下文谓词条件;

为所述适用的维度上下文谓词条件计算净收益度量;

基于为所述适用的维度上下文谓词条件计算的净收益度量,确定所述适用的维度上下文谓词条件将不与第一事实扫描操作相关联;以及

使用功能依赖性信息从所述适用的维度上下文谓词条件中推断第一维度上下文谓词条件。

10.如权利要求1或2所述的方法,其中事实记录的源是存储事实记录的表、物化的视图或在线分析处理(OLAP)索引。

11.如权利要求1或2所述的方法,另外其中识别第一维度上下文谓词条件包括:

识别原始查询计划中的第三操作,其中第一维度上下文谓词条件与第三操作相关联。

12.如权利要求11所述的方法,其中第三操作与维度表相关联。

13.如权利要求1或2所述的方法,其中识别第一维度上下文谓词条件包括:

识别要与第一事实扫描操作相关联的第二维度上下文谓词条件;以及

将第二维度上下文谓词条件转化成第一维度上下文谓词条件。

14.如权利要求13所述的方法,其中所述转化包括使用功能依赖性信息将第二维度上下文谓词条件转化成第一维度上下文谓词条件。

15.如权利要求13所述的方法,其中第二维度上下文谓词条件指定用于第一维度字段的值,并且第一维度上下文谓词条件指定用于与第一维度字段不同的第二维度字段的值,其中第二维度字段是维度表中或事实记录的源中的列字段。

16.如权利要求1或2所述的方法,还包括:

识别原始查询计划中的第一子计划,第一子计划包括第一事实扫描操作,其中第一子计划仅包括聚合、联接、投影、过滤或事实扫描操作中的一个或多个;以及

其中增强的查询计划包括具有与第一事实扫描操作相关联的第一维度上下文谓词条件的第一子计划。

17.如权利要求1或2所述的方法,还包括:

基于星形架构,确定事实记录的源能够与维度表联接;以及

其中第一维度上下文谓词条件包括与要与事实记录的源联接的维度表中的维度键相关的条件。

18.如权利要求1或2所述的方法,其中:

事实记录的源是OLAP索引,该OLAP索引包括表和对该表中的维度值加索引的索引;

该表包括通过将存储事实记录的事实表与包括维度值的维度表联接而产生的数据;以及

针对OLAP索引中的维度值来评估第一维度上下文谓词条件。

19.一种非暂态计算机可读存储器,存储能够由一个或多个处理器执行的多条指令,所述多条指令包括在由所述一个或多个处理器执行时使所述一个或多个处理器执行处理的指令,所述处理包括:

接收为用于查询关系数据库中存储的数据的查询而生成的原始查询计划,原始查询计划包括被配置为从事实记录的源扫描事实记录的第一事实扫描操作,原始查询计划还包括被配置为从第一事实扫描操作接收记录集的第二操作;

识别要与第一事实扫描操作相关联的第一维度上下文谓词条件,其中第一维度上下文谓词条件对应于增强的查询计划中的第一事实扫描操作,所述增强的查询计划被配置为扫描所述事实记录的源的一个或多个记录中的事实;以及

重写原始查询计划以生成增强的查询计划,其中,在增强的查询计划中,第一维度上下文谓词条件与第一事实扫描操作相关联,并且第二操作仅接收满足第一维度上下文谓词条件的那一个或多个被扫描的事实记录;

执行增强的查询计划以获得用于查询的结果记录集,其中,由于第一维度上下文谓词条件与第一事实扫描操作的关联,增强的查询计划比原始查询计划执行得快;以及

提供结果记录集作为对查询的响应。

20.一种用于生成增强的查询计划的系统,包括:

一个或多个处理器;以及

耦合到所述一个或多个处理器的存储器,所述存储器存储能够由所述一个或多个处理器执行的多条指令,所述多条指令包括在由所述一个或多个处理器执行时使所述一个或多个处理器执行处理的指令,所述处理包括:

接收为用于查询关系数据库中存储的数据的查询而生成的原始查询计划,原始查询计划包括被配置为从事实记录的源扫描事实记录的第一事实扫描操作,原始查询计划还包括被配置为从第一事实扫描操作接收记录集的第二操作;

识别要与第一事实扫描操作相关联的第一维度上下文谓词条件,其中所述第一维度上下文谓词条件对应于增强的查询计划中的第一实施扫描操作,所述增强的查询计划被配置为扫描所述事实记录的源的一个或多个记录中的事实;以及

重写原始查询计划以生成增强的查询计划,其中,在增强的查询计划中,第一维度上下文谓词条件与第一事实扫描操作相关联,并且第二操作仅接收满足第一维度上下文谓词条件的那一个或多个被扫描的事实记录;

执行增强的查询计划以获得用于查询的结果记录集,其中,由于第一维度上下文谓词条件与第一事实扫描操作的关联,增强的查询计划比原始查询计划执行得快;以及

提供结果记录集作为对查询的响应。

# 优化SQL查询计划的维度上下文传播技术

[0001] 对其它申请的引用

[0002] 本申请要求以下申请的利益和优先权,出于所有目的,通过引用将其全部内容并入本文:

[0003] (1)于2018年1月16日提交的标题为“DIMENSION CONTEXT PROPAGATION TECHNIQUE FOR ANALYTICAL SQL QUERIES”的美国临时申请No.62/617,970;

[0004] (2)于2018年10月18日提交的标题为“DIMENSION CONTEXT PROPAGATION TECHNIQUE FOR ANALYTICAL SQL QUERIES”的美国临时申请No.62/747,642;以及

[0005] (3)于2019年1月15日提交的标题为“DIMENSION CONTEXT PROPAGATION TECHNIQUES FOR OPTIMIZING SQL QUERY PLANS”的美国非临时申请No.16/248,061。

## 技术领域

[0006] 本公开一般而言涉及查询的执行。更具体地,描述了用于增强查询执行性能的技术。本文描述了各种发明性实施例,包括方法、系统、存储由一个或多个处理器可执行的程序、代码或指令的非暂态计算机可读存储介质。

## 背景技术

[0007] 分析方案使企业能够更好地理解其商业在诸如时间、空间、产品和客户之类的维度上的不同活动。这可以包括执行上下文或相关性分析、绩效管理、假设分析、预测、切片和切块分析、关键绩效指标(KPI)和趋势分析、仪表板等。任何多维分析的关键方面都是指定分析的重点和分析的输入两者的维度上下文。

[0008] 在线分析处理(OLAP)数据库系统常常被用于挖掘数据并生成报告,以便获得见解或提取关于各种与商业相关的活动的有用模式。例如,OLAP数据库可以被用于识别销售趋势、分析营销活动和预测财务绩效。这通常涉及针对大量(例如,数百万条记录)的多维数据库数据评估分析查询。例如,可以根据诸如时间、地理区域、部门和/或各个产品之类的维度来分析销售数据。

[0009] 分析人员一般关于其数据考虑到了语义模型。通常会在维度和度量方面考虑数据,其中度量捕获关于活动的详细信息(例如,销售额、销售数量),而维度捕获活动的上下文(例如,所售产品的类型、发生销售的商店、商店所在的州、购买的客户等)。商业数据一般是多维的并且语义丰富。

[0010] 大量的数据分析方案使用关系数据库系统作为后端,并且诸如SQL(结构化查询语言)之类的某些查询语言被用作分析语言。多维数据库数据通常包括存储度量的事实记录(例如,销售数据),该事实记录与存储维度或上下文信息的维度记录(例如,时间数据、地理数据、部门数据、产品数据)分开存储。例如,事实记录可以存储在事实表中,而维度记录可以存储在维度表中。

[0011] 在SQL领域中,分析方案建立在星形架构之上(因其实体-关系(ER)图的外观而得名,参见图4)并使用SQL查询进行查询。典型的分析查询可以涉及通过联接或联合而组合的

几个构造块查询子计划/公共表表达式。用于查询的查询子计划通常涉及将事实表与维度表联接并在多维空间的切片上进行聚合计算。执行查询的成本的大部分通常是事实记录的源的扫描。另外,根据维度记录来分析事实记录常常涉及将事实记录与维度记录联接,例如,将事实表与维度表联接。但是,将事实记录与维记录联接可能是计算密集的,这至少是因为它涉及处理大量事实记录。一些现有技术专注于通过使用预计算(诸如使用OLAP立方体、摘录和物化的预聚合的表)来优化查询执行。但是,由这些技术提供的执行改进仍然受到限制。此外,预聚合还有若干缺点,诸如粒度的丢失、无法运行自组织(ad-hoc)查询等。

[0012] 混合计算开销是其中常常起草和评估数据库查询的方式。数据库查询的起草一般外包给几乎没有领域或上下文知识的数据库程序员。另外,以诸如SQL之类的语言起草此类查询非常困难。查询的不同部分通常以隔离的方式被评估,并且这会导致不必要和重复的数据处理。例如,由于编写查询的方式,用于查询的查询计划可以在查询计划的后期阶段指定过滤操作。当在查询计划的早期阶段执行查询计划操作而不考虑过滤操作时,因为在早期阶段被处理的许多数据在后期阶段可能作为无关数据被过滤掉,所以在早期阶段执行的许多数据处理可能会变得毫无用处。

## 发明内容

[0013] 本公开一般而言涉及查询的执行。更具体地,描述了用于增强查询执行性能的技术。本文描述了各种发明性实施例,包括方法、系统、存储由一个或多个处理器可执行的程序、代码或指令的非暂态计算机可读存储介质。

[0014] 本公开一般而言涉及用于高效地执行诸如自组织数据库查询之类的分析查询的技术。对于诸如SQL自组织查询之类的查询,为该查询生成的查询计划经过优化和增强,并重写为经优化或增强的查询计划。增强的查询计划在被执行时比原始非优化的查询计划使用少的CPU周期,因此比原始查询计划执行得快。因此,为其生成增强的查询计划的查询执行得快,而不会损害获得的结果或正被查询的数据。在某些实施例中,优化或增强的查询计划在被执行时还比原始非优化的查询计划使用少的存储器资源。

[0015] 在某些实施例中,通过识别在原始查询计划中的一个或多个事实扫描操作/运算符的集合并且然后在重写的增强的查询计划中将维度上下文与事实扫描操作集合中的一个或多个相关联来优化原始查询计划。在某些实施例中,通过将与维度上下文对应的一个或多个谓词条件(也称为维度上下文谓词条件)与事实扫描操作相关联,将维度上下文与事实扫描操作相关联。事实扫描操作是查询计划中被配置为从事实记录的源(也称为事实源)扫描或读取事实(也称为事实记录或事实行)的操作。增强的查询计划中谓词条件与的事实扫描操作的关联也被称为谓词条件或维度上下文向事实扫描操作的传播或者谓词条件或维度上下文向事实源(事实扫描操作对该事实源进行操作)的传播。由于关联或传播,与原始查询计划相比,增强的查询计划中扫描/读取和/或处理事实记录的总成本要低得多。因此,增强的查询计划在执行时所使用的CPU周期比原始非优化的查询计划所使用的CPU周期少,因此比原始查询计划执行得快。

[0016] 在某些实施例中,优化查询计划的处理可以包括:识别查询计划中哪一个或多个事实扫描操作是谓词条件传播的候选,识别要与每个候选事实扫描操作相关联的一个或多个谓词条件,然后用该关联重写查询计划。用该关联执行查询的查询计划。本文公开的各种

不同技术可以被用于执行这个处理。

[0017] 可以使用各种不同的技术来识别要与事实扫描操作相关联的谓词条件。例如,来自原始查询计划的一个部分的谓词条件可以传播到重写的增强的查询计划中的查询计划的另一个部分中的事实扫描操作。作为示例,应用于查询计划的一个部分中的维度表的维度谓词条件可以被传播并与查询计划的另一个部分中的事实扫描操作相关联。在其它一些情况下,代替将特定谓词条件与事实扫描操作相关联,从查询计划中的特定谓词条件或原始谓词条件推断新谓词条件(也称为特定谓词条件或原始谓词条件到新谓词条件的转化或映射),并且新谓词条件与事实扫描操作而不是特定或谓词条件相关联。新谓词条件使得用新谓词条件执行事实扫描操作的成本(例如,执行的时间)小于用特定谓词条件或原始谓词条件执行事实扫描操作的成本。在一些情况下,代替从查询中引用的事实表中扫描事实记录,可以使用其它物理事实结构(事实源)(诸如OLAP索引、预聚合的物化的视图)来执行扫描。

[0018] 如上面所指示的,增强的查询计划在执行时比原始非优化的查询计划使用少的CPU周期,因此比原始查询计划执行得快。有多种方式可以实现这一点。在某些情况下,由于谓词条件与查询计划中的事实扫描操作的关联,与原始查询计划相比,由重写的查询计划处理的事实记录的数量减少了。例如,作为谓词条件与事实扫描操作相关联的结果,只有满足谓词条件的事实记录才从事实扫描操作提供给查询计划中的下一个下游操作。这减少了提供给增强的查询计划中的下游操作(例如,事实表和维度表之间的联接操作)并且必须由下游操作处理的事实记录的数量。这减少了与在增强的查询计划中执行下游操作相关联的计算开销。基于谓词条件对事实记录的这种过滤减少了查询计划对事实记录的不必要和重复的处理。由查询计划处理的事实记录数量的减少转化为查询计划以及为其生成查询计划的查询本身的更快执行时间。由查询计划处理的事实记录的数量的减少还可以转化为用于执行重写的查询计划的更少的存储器资源。如上所述,在某些实施例中,从特定谓词条件推断或转化的新谓词条件可以与事实扫描操作而不是与特定谓词条件相关联。新谓词条件使得用新谓词条件执行事实扫描操作的成本(例如,执行的时间)小于用特定谓词条件执行事实扫描操作的成本。例如,与特定谓词条件相关联的事实扫描操作可能花2秒来执行,而与新谓词条件相关联的相同事实扫描操作可能仅花1秒来执行。谓词条件的推断或转化使增强的查询计划比未优化的查询计划执行得快。此外,在某些实施例中,通过以这样的方式对可用的替代事实表示(诸如OLAP索引)执行事实扫描来实现更快的执行时间,以便充分利用事实表示的扫描能力。例如,转换成对某些谓词“P”的OLAP索引扫描,使得该组合可以充分利用快速谓词评估并跳过OLAP索引的扫描,从而在一些情况下导致事实扫描处理数量级减少。

[0019] 从性能的角度来看,与未增强的查询计划相比,增强的查询计划可以执行得更快,并且可以潜在使用更少的资源。基于本文公开的教导执行的大量查询比用未增强的查询计划的查询执行快10倍、25倍甚至100倍或甚至更高数量级。这是因为事实表比维度表大几个数量级,因此减少与处理事实记录相关联的扫描成本带来巨大的性能提升。例如,在一种情况下,自组织查询的查询执行时间从206秒减少到16秒。

[0020] 可以使用各种不同的信息片段来促进本文描述的查询计划优化。在某些实施例中,可以基于以下条件来优化和增强为数据库查询而生成的查询计划:查询的结构、为查询

生成的查询计划的结构、与正在被查询的数据库相关的模式信息以及可用于查询计划优化器和/或增强器的语义信息。例如,商业语义或商业智能(BI)语义信息可以被用于重写查询计划。语义信息可以包括具有丰富价值的商业数据和基于现实世界处理、组织结构等的结构语义。语义信息可以包括与商业模型有关的信息,如商业层次结构和数据中的功能依赖性。例如,依赖性信息可以包括描述数据库数据的不同字段(例如,列)之间的关系(例如,层次或非层次)的信息。例如,依赖性元数据可以包括描述“城市”列的值与“州”列的值之间的分层关系的信息(例如,旧金山是加利福尼亚州内的城市)。层次结构的示例包括时间段层次结构(例如,日、周、月、季度、年等)、地理层次结构(例如,城市、州、国家/地区)等。在某些实施例中,可以根据分析有关存储正被分析的数据的数据库的结构和模式的信息来确定语义信息,而无需任何用户输入。从这种分析得出的推断可以被用于重写查询计划,以生成优化和增强的查询计划。

[0021] 在某些实施例中,可以接收为用于查询存储在关系数据库中的数据的查询而生成的原始查询计划。原始查询计划可以包括被配置为从事实记录的源中扫描事实记录的第一事实扫描操作。原始查询计划还可以包括被配置为从第一扫描操作接收记录的集合的第二操作。识别要与第一事实扫描操作相关联的第一谓词条件。原始查询计划可以被重写以生成增强的查询计划,其中,在增强的查询计划中,第一谓词条件与第一扫描操作相关联,并且第二操作仅接收满足该谓词条件的那一个或多个被扫描的事实记录。由于第一谓词条件与第一扫描操作的关联,增强的查询计划比原始查询计划执行得快。可以执行增强的查询计划以获得用于查询的结果记录集。结果记录集可以作为对查询的响应被提供。

[0022] 由该查询正在查询的数据(或查询对其进行操作的数据)可以是非预聚合的数据。分析平台使得能够对此类非预聚合的数据进行自组织交互式查询。

[0023] 如上面所指示的,增强的查询计划比原始查询计划执行得快。这是因为与原始查询计划相比,增强的查询计划执行所花的CPU周期少。在一些实施例中,这是因为在增强的查询计划中,由第二操作接收并由第二操作处理的事实记录的数量小于原始查询计划中由第二操作接收并由第二操作处理的的事实记录的数量。

[0024] 在某些实施例中,识别第一谓词条件涉及将第一事实扫描操作识别为对原始查询计划中的第一事实表进行操作。从第一扫描操作开始,然后遍历原始查询计划,以识别第一事实扫描操作的一个或多个适用的谓词条件的列表,适用的谓词条件的所述列表包括第一谓词条件。

[0025] 在某些实施例中,识别第一谓词条件包括从原始查询计划中识别:第一事实扫描操作对第一事实表进行操作;第二事实扫描操作对第二事实表进行操作;第二事实表与维度表之间的联接操作,其中第一谓词条件与维度表相关联;以及第一事实表与第二事实表之间的公共维度,其中第一谓词条件基于来自该公共维度的属性。

[0026] 在某些实施例中,识别第一谓词条件包括识别适用于第一事实扫描操作的多个谓词条件。然后,针对多个谓词条件中的每个谓词条件计算净收益度量,其中针对多个谓词条件中的特定谓词条件的净收益度量是处理来自事实记录的源中的事实行的成本减去将该特定谓词条件应用于第一事实扫描操作的成本的减少的度量。然后,基于为多个谓词条件计算的净收益度量,对多个谓词条件中的谓词条件进行排序。然后基于该排序选择一个或多个谓词条件与第一事实扫描操作相关联。在某些实施例中,选择具有最高净收益度量的

谓词条件以与第一事实扫描操作相关联。

[0027] 在某些实施例中,识别第一谓词条件包括识别用于第一事实扫描操作的适用的谓词条件。计算适用的谓词条件的净收益度量。基于为适用的谓词条件计算的净收益度量,确定适用的谓词条件对于与第一事实扫描操作相关联是不可行的。然后使用功能依赖性信息从适用的谓词条件中推断要与第一事实扫描操作相关联的另一个谓词条件。

[0028] 事实记录的源可以是多种类型,包括存储事实记录的表、物化的视图或在线分析处理(OLAP)索引。

[0029] 在某些实施例中,识别第一谓词条件包括识别原始查询计划中的第三操作,其中第一谓词条件与第三操作相关联。第三操作可以与原始查询计划中的维度表相关联。

[0030] 在某些实施例中,识别第一谓词条件包括识别要与第一扫描操作相关联的第二谓词条件,以及将第二谓词条件转化成第一谓词条件。转化可以包括使用功能依赖性信息将第二谓词条件转化成第一谓词条件。在一个实例中,第二谓词条件指定用于第一维度字段的值,并且第一谓词条件指定用于与第一维度字段不同的第二维度字段的值,其中第二维度字段是维度表中或事实记录的源中的列字段。

[0031] 在某些实施例中,可以识别原始查询计划中的第一子计划,该第一子计划包括第一事实扫描操作,其中该第一子计划仅包括聚合、联接、投影、过滤或事实扫描操作中的一个或多个。增强的查询计划在被生成时可以包括具有与第一扫描操作相关联的第一谓词条件的第一子计划。

[0032] 在某些实施例中,该处理可以包括基于星形架构来确定事实记录的源能够与维度表联接,并且其中第一谓词条件包括与要与事实记录的源联接的维度表中的维度键相关的条件。

[0033] 在某些实施例中,事实记录的源是OLAP索引,该OLAP索引包括表和对该表中的维度值加索引的索引。表可以包括通过将存储事实记录的事实表与包括维度值的维度表联接而产生的数据,并且针对OLAP索引中的维度值来评估第一谓词条件。

[0034] 通过参考以下说明书、权利要求书和附图,前述以及其它特征和实施例将变得更加显而易见。

## 附图说明

[0035] 图1A和图1B是根据某些实施例的分析平台的简化框图。

[0036] 图2描绘了根据某些实施例的事实和维度表的集合。

[0037] 图3描绘了根据某些实施例的查询计划。

[0038] 图4描绘了根据某些实施例的星形架构。

[0039] 图5描绘了根据某些实施例的重写的查询执行计划。

[0040] 图6描绘了根据某些实施例的用于执行半联接操作的方法。

[0041] 图7描绘了根据某些实施例的在线分析处理(OLAP)索引。

[0042] 图8A和图8B描绘了根据某些实施例的用于执行索引半联接操作的方法。

[0043] 图9描绘了根据某些实施例的基于功能依赖性转化谓词条件的方法。

[0044] 图10A和图10B描绘了根据某些实施例的用于将谓词条件从维度记录的第一集合跨维度记录的第二集合传播到事实记录的集合的方法。

[0045] 图11描绘了根据某些实施例的包括多个事实表的架构。

[0046] 图12A-图12C描绘了根据某些实施例的用于将谓词条件跨维度记录的集合(来自维度表DT)跨事实记录的第一集合(事实源1“FS1”)传播到事实记录的第二集合(事实源2“FS2”)的方法。

[0047] 图13A是描绘根据某些实施例的生成增强的查询计划的方法的简化流程图。

[0048] 图13B是描绘根据某些实施例的与生成增强的查询计划的方法相关的更多细节的简化流程图。

[0049] 图14描绘了用于实现实施例的分布式系统的简化图。

[0050] 图15是根据某些实施例的基于云的系统环境的简化框图。

[0051] 图16图示了可以用于实现某些实施例的计算机系统。

[0052] 图17描绘了根据某些实施例的另一种架构。

## 具体实施方式

[0053] 在下面的描述中,出于解释的目的,阐述了具体细节以便提供对某些发明实施例的透彻理解。但是,将明显的是,可以在没有这些具体细节的情况下实践各种实施例。附图和描述不意图是限制性的。

[0054] 本公开一般而言涉及用于高效执行诸如交互式自组织数据库查询之类的分析查询的技术。对于诸如SQL自组织查询之类的查询,为查询而生成的查询计划经过优化和增强,并被重写为经优化的或增强的查询计划。增强的查询计划在被执行时使用比原始非优化的查询计划少的CPU周期,因此比原始查询计划执行得快。因此,为其生成增强的查询计划的查询执行得更快,而不会损害获得的结果或正被查询的数据。在某些实施例中,优化或增强的查询计划在被执行时还使用比原始非优化的查询计划少的存储器资源。

[0055] 在某些实施例中,通过识别原始查询计划中的一个或多个事实扫描操作/运算符的集合并且然后在重写的增强的查询计划中将维度上下文与事实扫描操作集合中的一个或多个相关联来优化原始查询计划。在某些实施例中,通过将与维度上下文对应的一个或多个谓词条件(也称为维度上下文谓词条件)与事实扫描操作相关联,将维度上下文与事实扫描操作相关联。事实扫描操作是查询计划中被配置为从事实记录的源(也称为事实源)扫描或读取事实(也称为事实记录或事实行)的操作。增强的查询计划中谓词条件与的事实扫描操作的关联也被称为谓词条件或维度上下文向事实扫描操作的传播或者谓词条件或维度上下文向事实源(事实扫描操作在该事实源上操作)的传播。由于关联或传播,与原始查询计划相比,增强的查询计划中扫描/读取和/或处理事实记录的总成本要低得多。因此,增强的查询计划在执行时所使用的CPU周期比原始非优化的查询计划所使用的CPU周期少,因此比原始查询计划执行得快。

[0056] 在某些实施例中,优化查询计划的处理可以包括:识别查询计划中哪一个或多个事实扫描操作是谓词条件传播的候选、识别要与每个候选事实扫描操作相关联的一个或多个谓词条件,然后用该关联重写查询计划。用该关联执行查询的查询计划。本文公开的各种不同技术可以被用于执行这个处理。

[0057] 可以使用各种不同的技术来识别要与事实扫描操作相关联的谓词条件。例如,来自原始查询计划的一个部分的谓词条件可以传播到重写的增强的查询计划中的查询计划

的另一个部分中的事实扫描操作。作为示例,应用于查询计划的一个部分中的维度表的维度谓词条件可以被传播并与查询计划的另一个部分中的事实扫描操作相关联。在其它一些情况下,代替将特定谓词条件与事实扫描操作相关联,从查询计划中的特定谓词条件或原始谓词条件推断新谓词条件(也称为特定谓词条件或原始谓词条件到新谓词条件的转化或映射),并且新谓词条件与事实扫描操作而不是特定或谓词条件相关联。新谓词条件使得用新谓词条件执行事实扫描操作的成本(例如,执行的时间)小于用特定谓词条件或原始谓词条件执行事实扫描操作的成本。在一些情况下,代替从查询中引用的事实表中扫描事实记录,可以使用其它物理事实结构(事实源)(诸如OLAP索引、预聚合的物化的视图)来执行扫描。

[0058] 如上面所指示的,增强的查询计划在执行时使用比原始非优化的查询计划少的CPU周期,因此比原始查询计划执行得更快。有多种方式可以实现这一点。在某些情况下,由于谓词条件与查询计划中的事实扫描操作的关联,与原始查询计划相比,由重写的查询计划处理的事实记录的数量减少了。例如,作为谓词条件与事实扫描操作相关联的结果,只有满足谓词条件的事实记录才从事实扫描操作提供给查询计划中的下一个下游操作。这减少了提供给增强的查询计划中的下游操作(例如,事实表和维度表之间的联接操作)并且必须由下游操作处理的事实记录的数量。这减少了与在增强的查询计划中执行下游操作相关联的计算开销。基于谓词条件对事实记录的这种过滤减少了查询计划对事实记录的不必要和重复的处理。由查询计划处理的事实记录数量的减少转化为查询计划以及为其生成查询计划的查询本身的更快执行时间。由查询计划处理的事实记录的数量的减少还可以转化为用于执行重写的查询计划的更少的存储器资源。如上所述,在某些实施例中,从特定谓词条件推断或转化的新谓词条件可以与事实扫描操作而不是与特定谓词条件相关联。新谓词条件使得用新谓词条件执行事实扫描操作的成本(例如,执行的时间)小于用特定谓词条件执行事实扫描操作的成本。例如,与特定谓词条件相关联的事实扫描操作可能花2秒来执行,而与新谓词条件相关联的相同事实扫描操作可能仅花1秒来执行。谓词条件的推断或转化使增强的查询计划比未优化的查询计划执行得快。此外,在某些实施例中,通过以这样的方式对可用的替代事实表示(诸如OLAP索引)执行事实扫描来实现更快的执行时间,以便充分利用事实表示的扫描能力。例如,转换成对某些谓词“P”的OLAP索引扫描,使得该组合可以充分利用快速谓词评估并跳过OLAP索引的扫描,从而在一些情况下导致事实扫描处理数量级减少。

[0059] 从性能的角度来看,与未增强的查询计划相比,增强的查询计划可以执行得更快,并且可以潜在使用更少的资源。基于本文公开的教导执行的大量查询比用未增强的查询计划的查询执行快10倍、25倍甚至100倍或甚至更高数量级。这是因为事实表比维度表大几个数量级,因此减少与处理事实记录相关联的扫描成本带来巨大的性能提升。例如,在一种情况下,自组织查询的查询执行时间从206秒减少到16秒。查询可以在几秒钟之内对大型数据集(例如,TB级数据集)执行。

[0060] 可以使用各种不同的信息片段来促进本文描述的查询计划优化。在某些实施例中,可以基于以下条件来优化和增强为数据库查询而生成的查询计划:查询的结构、为查询生成的查询计划的结构、与正在被查询的数据库相关的模式信息以及可用于查询计划优化器和/或增强器的语义信息。例如,商业语义或商业智能(BI)语义信息可以被用于重写查询

计划。语义信息可以包括具有丰富价值的商业数据和基于现实世界处理、组织结构等的结构语义。语义信息可以包括与商业模型有关的信息，如商业层次结构和数据中的功能依赖性。例如，依赖性信息可以包括描述数据库数据的不同字段(例如，列)之间的关系(例如，层次或非层次)的信息。例如，依赖性元数据可以包括描述“城市”列的值与“州”列的值之间的分层关系的信息(例如，旧金山是加利福尼亚州内的城市)。层次结构的示例包括时间段层次结构(例如，日、周、月、季度、年等)、地理层次结构(例如，城市、州、国家/地区)等。语义信息可以包括数据立方体模型语义信息。在某些实施例中，用于查询计划增强的元数据可以包括关于如何物理存储被分析的数据的信息，还有识别数据之间的关系的逻辑信息。在某些实施例中，可以根据分析有关存储正被分析的数据的数据库的结构和模式的信息来确定语义信息，而无需任何用户输入。从这种分析得出的推断可以被用于重写查询计划，以生成优化和增强的查询计划。

[0061] 例如，可以分析为查询生成的查询计划以识别对事实源的事实扫描操作。然后可以使用语义信息(例如，事实和维度上下文表的联接操作、数据库列之间的功能依赖性、事实的物理物化)在重写的优化的查询计划中主动识别和推动关于事实扫描操作的维度上下文条件。这导致在重写的查询计划中处理事实记录所涉及的成本(例如，时间)的显著减少。因此，与未优化的查询计划相比，重写的查询计划执行得更快(例如，使用更少的CPU周期)，并且在许多情况下，可以使用更少的存储器资源。这转化为为其生成查询计划的查询执行得更快，并且潜在地使用更少的存储器资源。

[0062] 图1A是结合了示例实施例的分析平台或基础设施100的简化框图。分析平台100可以包括经由一个或多个通信网络彼此通信耦合的多个系统。图1A中的系统包括经由一个或多个通信网络彼此通信耦合的处理系统150和存储系统106。通信网络可以促进图1A中所描绘的各种系统之间的通信。该通信网络可以是各种类型，并且可以包括一个或多个通信网络。这种通信网络的示例包括但不限于互联网、广域网(WAN)、局域网(LAN)、以太网、公共或专用网络、有线网络、无线网络等，及其组合。可以使用不同的通信协议来促进通信，包括有线和无线协议，诸如IEEE 802.XX协议套件、TCP/IP、IPX、SAN、**AppleTalk®、Bluetooth®**和其它协议。通信网络可以包括促进图1A中所描绘的各种系统之间的通信的任何基础设施。

[0063] 图1A中描绘的分析平台100仅仅是示例，并且无意于不当地限制所要求保护的实施例的范围。平台可以根据需要弹性地向外扩展，以适应正被查询的数据尺寸的改变和工作负载的改变(例如，正在被并行或以其它方式执行的查询的数量)。本领域普通技术人员将认识到许多可能的变化、替代和修改。例如，在一些实施方式中，分析平台100可以具有比图1A中所示的更多或更少的系统或组件，可以组合两个或更多个系统，或者可以具有不同的系统配置或布置。另外，图1A中描绘并在本文描述的基础设施可以在包括独立或集群实施例的各种不同环境中实现为内部部署或云环境(可以是包括私有、公共和混合云环境的各种类型的云)、内部部署环境、混合环境等。

[0064] 在某些实施例中，待分析或查询的数据可以由存储系统106存储。存储在存储系统106中的数据可以来自一个或多个数据源。可以以各种形式(诸如对象存储、块存储、仅一堆盘(jbod)、如在计算节点上的盘中等等以及它们的组合)来组织和存储数据。存储系统106可以包括用于存储数据的易失性存储器和/或非易失性存储器。例如，存储系统106可以包

括存储待分析的商业数据的数据库。商业数据一般是多维的,并且语义丰富。存储系统106表示分析平台100的存储层,其包括持久存储数据的物理存储组件(例如,旋转硬盘、SSD、存储器高速缓存等),并且还包含以最佳地服务于具体查询工作负载的方式提供无数数据结构用于存储关系/空间/图形数据的软件。存储系统106中的数据不必全部位于一个位置,而是可以以分布式方式存储。可以将数据组织和存储在各种存储/计算环境中,诸如数据湖、数据仓库、Hadoop集群等。

[0065] 存储系统106和处理系统150的存储器资源可以包括系统存储器和非易失性存储器。系统存储器可以为一个或多个处理器提供存储器资源。系统存储器通常是易失性随机存取存储器(RAM)(例如,动态随机存取存储器(DRAM)、同步DRAM(SDRAM)、双倍数据速率SDRAM(DDR SDRAM))的形式。与操作系统和由一个或多个处理器执行的应用或处理相关的信息可以存储在系统存储器中。例如,在运行时期间,可以将操作系统/内核加载到系统存储器中。此外,在运行时期间,与由服务器计算机执行的一个或多个应用相关的数据可以被加载到系统存储器中。例如,由服务器计算机执行的应用可以被加载到系统存储器中并且由一个或多个处理器执行。服务器计算机可以能够并行执行多个应用。

[0066] 非易失性存储器可以被用于存储将被持久存储的数据。非易失性存储器可以以不同的形式出现,诸如硬盘、软盘、闪存、固态驱动器或磁盘(SSD)、USB闪存驱动器、存储卡、存储棒、盒式磁带、zip盒、计算机硬盘驱动器、CD、DVD、网络附接存储装置(NAS)、经由存储区域网络(SAN)提供的存储器存储装置等。在某些情况下,当将应用部署到服务器计算机或安装在服务器计算机上时,与应用相关的信息可以存储在非易失性存储器中。

[0067] 在某些实施例中,存储系统106中的数据可以在关系数据库中被存储在逻辑上被结构化为一个或多个星形架构的表中,或者存储在具有专用存储结构(诸如多维阵列(通常也称为多维立方体))的系统中。星形架构包括引用任何数量的维度表的中央事实表。每个事实表以事实记录的形式存储事实或度量。事实表可以链接到将事实与其上下文相关联的维度表。维度表存储描述事实表行中存储的度量数据的上下文信息。由于事实表通常与许多维度相关联,因此该结构在图形上看起来像星形,因此其名称由此而来。事实的数量与唯一维度值的数量之间一般存在一个数量级差异,其中事实在数目上远远超过维度。例如,企业可以有数亿个销售事实记录,但是只有少数几个(例如,几千个)维度(诸如商店、销售的州等)。因此,事实表的尺寸一般非常大,并且比维度表大一个数量级。

[0068] 这里,术语“星形架构”以其最一般的形式使用,以捕获事实行与许多维度表行之间的逻辑关系;因此,如雪花架构之类的变体也被隐含并被包括在本公开中所使用的术语“星形架构”下。雪花架构与星形架构的区别在于事实行和维度之间的关系可以是间接的(要求多于直接关联,例如销售事实行链接到客户行,而客户行链接到客户地址行)。

[0069] 查看分析数据的另一种方式是作为n维立方体。典型的分析着眼于这个庞大的多维空间的某个任意子空间(也称为切片)。任何特定分析的重点都可以跨越多个维度和活动,可以具有多个步骤,涉及实体及其活动之间的任意链接。在关系世界中,数据立方体可以被建模为星形架构。

[0070] 在关系数据库中,事件或交易(例如,产品的销售)可以在联接到许多维度表的大型事实表中捕获。分析方案可以包括许多这样的星形架构,其中事实表具有共同的维度集合。图4描绘了与和商店退货的交易相关的数据对应的示例星形架构400。架构400捕获商店

退货交易并描述事实记录与维度记录之间的一对多关系。根据图4中所描绘的架构400，Store_Returns是存储事实记录的事实表(使用强调的边界示出)，其包含与商店退货交易相关的度量信息，而其它表(日期(Date)、商店(Store)、货物(Item)、客户(Customer)、Customer_Address)是存储事实记录的上下文信息的维度表。记录每个退货交易及相关联的上下文，包括关于退货商店的信息(包括商店标识符、城市和州)、关于退货客户的信息以及交易的时间信息(月、日、年)。星形架构400指示商店退货是按Date、Customer、Item、Store和Customer_Address维度记录来确定维度的，并且可以与Store_Returns事实记录联接。

[0071] 图2描绘了示例数据库数据200，其包括事实表202和多个维度表204-208，每个维度表与和图4中描绘的星形架构400对应的事实表202相关。事实表202包括外键列210、212和214，其可以分别与表204中的主键列218、表206中的226和表208中的230联接，以向事实表202添加列而不是行。虽然图2将数据库数据200描绘为表，但是应当认识到的是，数据库数据200可以被表示为索引、物化的视图和/或任何其它元组集。

[0072] 在图2的示例中，事实表202还包括存储事实度量列216(return_amt)的事实记录。外键列210、212和214使事实度量列216能够根据一个或多个维度列(例如，维度列220、222、224、228、232和234)来被分析。一般而言，如图2中所描绘的，事实记录通常比维度记录多得多。例如，事实记录可以数以百万计，而维度记录可以数以千计。

[0073] 处理系统150可以被配置为使得能够分析由存储系统106存储的数据。可以使用诸如SQL查询之类的分析查询来执行分析。处理系统150可以被配置为接收一个或多个查询130。在某些用例中，查询130可以由用户使用由处理系统150提供的界面来输入。在其它情况下，查询130可以由处理系统150从其它应用、系统或设备接收。在一些用例中，处理系统150可以生成其自己的查询。

[0074] 在某些实施例中，对于查询，处理系统150被配置为生成用于查询的增强的查询计划。下面更详细地描述的各种不同技术可以被用于生成增强的查询计划。然后，处理系统150可以针对由存储系统106存储的数据执行增强的查询计划，并生成结果集。然后，处理系统150可以输出结果集，作为对接收到的查询的响应。

[0075] 如在图1A中所描绘的实施例中那样，可以提供OLAP加索引126(例如，OLAP索引)以非常快速地从关系数据库中的事实源扫描/读取事实。在某些实施例中，处理系统150可以被配置为生成一个或多个OLAP索引126。

[0076] 在某些实施例中，图1A中描绘的分析平台100可以在一个或多个计算机的集群上运行。例如，可以使用由Apache Spark或Apache Hadoop提供的分布式计算框架来实现分析平台100。每个计算机可以包括一个或多个处理器(或中央处理单元(CPU))、存储器、存储设备、网络接口、I/O设备等。图16中描绘了这种计算机的示例，并在下面进行描述。一个或多个处理器可以包括单核或多核处理器。一个或多个处理器的示例可以包括但不限于通用微处理器，诸如由**Intel®、AMD®、ARM®**、Freescale半导体公司等提供的通用微处理器，其在被存储在相关联的存储器中的软件的控制下操作。分析平台100的软件组件可以作为应用在操作系统之上运行。由服务器计算机执行的应用可以由一个或多个处理器执行。

[0077] 作为由处理系统150生成的增强的查询计划的结果，使得可以更快地执行为其生

成增强的查询计划的查询。这转化为针对查询的更快响应时间。这些快速的响应时间使得能够以交互方式分析存储在存储系统106中的多维数据,而不会损害结果或被分析的数据。例如,可以针对与机器学习技术、点击流和事件/时间序列相关联的数据,并且更一般而言针对规模和复杂性可以快速增长的任何数据集,运行交互式查询。

[0078] 另外,查询直接在存储系统106中数据库中存储的数据上运行,而不是在任何预聚合的数据上运行。因此,查询可以就地在大型数据集上运行,而无需创建预聚合的数据或摘录。代替地,可以对实际的原始未加工数据本身执行查询。在某些实施例中,如本文所述,通过使用谓词条件并且使用存储器内OLAP索引与完全分布式计算引擎,可以使查询更快。这使得分析平台100能够促进自组织查询,该自组织查询不能在预聚合的数据上运行。自组织查询是未预定义的查询。一般创建自组织查询以在信息变得可用时分析信息。本文描述的分析平台100提供了以非常成本有效的方式对非常大的(兆兆字节及以上)多维数据(例如,存储在数据湖中的数据)运行交互式自组织查询的能力;过去使用常规技术对非常大(兆兆字节及以上)的多维数据运行这样的交互式自组织查询非常昂贵,而且无法扩展。

[0079] 另外,使用Apache Spark的分析平台的实施方式本身就使查询得以大规模执行。提供了弹性环境,其中计算机和存储装置(例如,可用于分析的新数据)可以独立地扩展。

[0080] 在分析平台100中,待查询的数据存储在中央位置中,可以使用一种或多种不同的分析工具从该位置分析该数据。不需要通过不同工具进行数据的独特准备。不同用户可以用他们选择的工具访问数据,这些工具包括运行Python或R的Zeppelin或Jupyter笔记本、如Tableau的BI工具等。

[0081] 对于大量的多维数据,难以大规模执行交互式查询。这是因为交互式查询要求快速响应时间。对于大量的多维数据,查询性能随着数据量(例如,兆兆字节的数据)的增加以及同时尝试访问数据集的用户数量的增加而降低。另外,事实表和维度表之间的联接会造成附加的性能瓶颈。过去,预聚合技术已用于缓解这个问题。例如,已使用OLAP立方体、摘录和物化的预聚合的表来促进多维数据的分析。

[0082] 预聚合的数据(也称为聚合的数据或聚合物)是已经使用一种或多种预聚合(或聚合)技术从一些底层数据(称为原始数据或未加工的数据)生成的数据。数据聚合或预聚合技术包括以汇总形式表示数据的任何处理。预聚合的数据可以包括预先计算或汇总的数据,并且可以存储在预聚合的表、摘录、OLAP立方体等中。当聚合事实时,通过消除维度或者通过将事实与积累的维度相关联来完成。因此,预聚合的数据不包括来自从中生成预聚合的数据的原始或未加工的数据的所有细节。

[0083] 但是,预聚合具有若干缺点。所有预聚合技术都会导致粒度损失。由于仅提供某些预定的预聚合,因此预聚合导致数据的细节的丢失。例如,原始或未加工的数据集可以每天记录销售信息。如果然后基于该数据执行基于每月的预聚合,那么每天或每日的信息丢失。为所有维度和度量的所有组合构建预聚合数据集或多维数据集是不切实际的。因此,预聚合限制了执行自组织查询的能力,因为更高级别的预聚合汇总(例如,每月信息)后面的关键信息(例如,每日信息)不可用。查询是针对预聚合的数据运行的,而不是对从中生成预聚合的原始或未加工的数据运行。因此,聚合改变了可以执行分析的粒度。事实的粒度是指记录事件的最低级别。例如,产品的销售可以记录有销售时间和时间戳。销售的聚合可以跨多个级别,诸如按渠道和产品的收入或按类别、产品、商店、州等的收入。按多个粒度和维度浏

览数据的能力是重要的。用于预聚合的正确粒度难以预先预计。因此,在涉及大数据分析时,预聚合的立方体和物化的汇总表会崩溃。

[0084] 分析平台100避免与基于预聚合的分析技术相关联的各种弊端。分析平台100提供了用于按规模和速度以不同粒度和维度执行多维数据的数据分析的框架。

[0085] 为了本申请的目的,术语“非预聚合的数据”用于指不包括预聚合的或预计算的数据的数据。在某些实施例中,分析平台100使得能够对存储在存储系统106中的非预聚合的数据执行查询。这使得能够对存储在存储系统106中的非预聚合的数据执行自组织查询。此外,由于处理系统150所提供的查询计划增强,因此查询的响应时间大大减少,从而使交互式查询得以运行。本文描述的分析平台100提供了以非常成本有效的方式对非常大的(兆兆字节及以上)多维数据(例如,存储在数据湖中的数据)运行交互式自组织查询的能力;过去使用常规技术对非常大(兆兆字节及以上)的多维数据运行这样的交互式自组织查询成本非常高,而且无法扩展。

[0086] 图1B是示出根据某些实施例的图1A中描绘的分析平台100的更详细视图的简化框图。图1B中描绘的分析平台100仅仅是示例,并且不旨在进行限制。本领域普通技术人员将认识到许多可能的变化、替代和修改。例如,在一些实施方式中,数据分析平台100可以具有比图1B中所示的更多或更少的组件、可以组合两个或更多个组件,或者可以具有不同的组件配置或布置。另外,图1B中所描绘并且本文描述的基础设施可以在各种不同的环境(包括独立的实施例、云环境(可以是各种类型的云,包括私有、公共和混合云环境)、内部部署环境、混合环境等)中实现。

[0087] 在图1B的示例中,图1A中所描绘的处理系统150包括数据分析系统102和数据库管理系统(DBMS)104,它们可以彼此通信耦合并且还耦合到存储系统106。在一些实施例中,数据分析系统102、DBMS104和存储系统106可以表示OLAP数据库系统的不同逻辑层。在一些实施例中,数据分析系统102、DBMS104和存储系统106可以是物理上分开的系统。在其它实施例中,它们可以组合成一个系统。DBMS104和数据分析系统102一起提供了强大的分析平台,该平台将商业智能技术与Apache Spark的能力相结合。

[0088] 存储系统106可以在事实和维度表中存储要分析的数据。如图1B中所示,可以有多个事实记录的源108,包括事实表124、OLAP索引126、物化的视图128和/或其它源。事实表124可以包括一个或多个行和一个或多个列,其中每一行可以表示事实记录,而列可以表示记录的字段。在一些实施例中,存储系统106可以在分析平台100内分布和/或虚拟化。

[0089] 存储系统106还可以将与事实记录相关联的上下文存储为存储一个或多个维度记录的一个或多个维度表110。通常,事实记录存储可以根据维度表110中的维度记录进行分析的度量的值。例如,事实记录可以包括销售额,而维度记录可以包括日期值,根据这些日期值可以聚合、过滤或以其它方式分析销售额。事实记录源108(例如,事实表、OLAP索引和物化的视图)以及维度表110和记录将在下面更详细地讨论。

[0090] 存储系统可以存储元数据109。元数据109可以包括关于关系工件(诸如表、列、视图等)的信息。

[0091] 在图1B的示例中,存储系统106可以附加地存储与数据库管理系统相关联的其他信息片段,诸如架构信息116和统计信息112。架构信息116可以识别与数据库相关联的表、索引和其它工件的结构。例如,对于星形架构,架构信息116可以包括描述事实记录与维度

记录之间的关系的信息。例如,架构信息116可以被用于确定可以根据时间维度和地理区域维度来查询销售数据。在某些实施例中,架构信息116可以由领域专家或分析师定义。架构信息116可以定义星形架构,诸如图4和图9中所描绘的模式。

[0092] 统计信息112可以包括关于事实记录或维度记录的信息,以及与查询计划和数据库管理系统104的性能相关属性相关的其它信息。例如,统计数据112可以包括表级统计信息(例如,表中的行数、用于表的数据块数、或表中的平均行长度)和/或列级统计信息(例如,列中不同值的数量或在列中找到的最小值/最大值)。

[0093] DBMS104可以被配置为(例如,包括软件,当由一个或多个处理器执行时,该软件启用功能)提供使得能够创建一个或多个数据库、将数据添加到数据库、更新和/或删除存储在数据库中的数据以及与管理数据库和存储的数据相关的其它功能。DBMS104还可以被配置为接收一个或多个查询130以分析存储在存储系统106中的数据、执行与查询对应的分析,并输出查询结果132。查询130可以采用不同的形式和语言,诸如结构化查询语言(SQL)。

[0094] 可以从诸如DBMS104的用户之类的一个或多个源、从另一个系统等接收查询130。例如,在图1B中所描绘的实施例中,在某些情况下,可以从数据分析系统(DAS)102接收查询。DAS102可以被配置为执行对存储在存储系统106中的数据的分析。DAS102可以包括被配置为生成分析查询以供DBMS104对由存储系统106存储的数据执行的软件。例如,如果存储系统106存储销售记录,那么DAS102可以生成用于分析销售记录的查询(例如,用于分析销售趋势的查询、用于分析广告活动的查询、以及对获得关于商业活动的见解或摘录有用模式的其它查询)并将该查询发送到DBMS104。在某些实施例中,DAS102可以使用存储在存储系统106中的信息(例如,架构信息116)来生成分析查询。例如,星形架构116可以被用于确定可以根据时间维度和地理区域维度来查询销售数据。由DBMS104检索到的与从DAS102接收到的查询对应的结果可以由DBMS104提供给DAS102。

[0095] 如图1B中所描绘的,DBMS104可提供用于接收查询130的界面122。解析器120可以被配置为对接收到的查询执行语法和语义分析。例如,对于SQL查询,解析器120可以被配置为检查查询中的SQL语句以获取正确的语法,并检查查询中引用的数据库对象和其它对象属性是否正确。

[0096] 在查询已经通过由解析器120执行的语法和语义检查之后,优化器114可以被配置为确定查询计划,以高效地执行查询。优化器114可以输出用于最优地执行查询的查询计划。在某些实施例中,执行计划包括要被执行的操作/运算符的系列或管线。执行计划可以由带根的树或图表示,其中树或图的节点表示各个操作/运算符。查询计划图一般是从图的叶子开始并朝着根前进来执行的。查询计划的叶子通常表示对事实记录或行进行扫描或从事实源(例如,事实表)中读取事实记录或行的事实扫描操作。查询计划中任何关系运算符的输出都是计算出的数据集。由查询计划中的操作返回的计算出的数据集(例如,行或记录的集合)被提供为查询计划管线中的下一个操作(也称为父操作或下游操作)的输入并由其消耗。在查询计划管线中的最后一个操作(查询计划图的根)中,事实行作为SQL查询的结果被返回。因此,查询计划中特定操作所返回的行成为查询计划管线中下一个操作(该特定操作的父操作或下游操作)的输入行。由操作返回的行的集合有时被称为行集合,而在查询计划中生成行集合的节点被称为行源。因此,查询计划表示从一个操作到另一个操作的行源的流。查询计划的每个操作从数据库中检索行,或者接受来自一个或多个其它子操作或上

游操作(或行源)的行。

[0097] 在某些实施例中,优化器114可以执行基于成本的优化以生成查询计划。在某些实施例中,优化器114可以努力为查询生成查询计划,该查询计划能够使用包括I/O、CPU资源和存储器资源在内的最少数量的资源来执行查询(最大吞吐量)。优化器114可以使用各种信息片段(诸如关于事实源108的信息、架构信息116、提示和统计信息112)来生成查询计划。在某些实施例中,优化器114可以生成潜在计划的集合并估计每个计划的成本,其中计划的成本是与执行该计划所需的预期资源使用成比例的估计值。计划的成本可以基于访问路径、计划中的操作(例如,联接操作)。访问路径是从数据库检索数据的方式。事实表中的行或记录可以通过全表扫描(例如,该扫描读取事实表中的所有行/记录并过滤掉不满足选择准则的行/记录)、使用索引(例如,通过遍历索引、使用加索引的列值来检索行)、使用物化的视图或rowid扫描来检索。然后,优化器114可以比较计划的成本并选择成本最低的计划。

[0098] 在某些实施例中,增强器103被配置为进一步增强/优化由优化器114选择的查询计划并生成增强的优化的查询计划。由增强器103生成的增强的查询计划比由优化器114生成的查询计划更好,因为它比未增强的查询计划执行得更快,并且在许多情况下,可以使用比未增强的查询计划更少的资源。

[0099] 然后,可以将由增强器103生成的优化的增强的查询计划提供给执行器子系统105,该执行器子系统105被配置为针对存储在存储系统106中的数据执行查询计划。然后,通过执行器105执行从增强器103接收到的查询计划而获得的结果可以由DBMS104返回,作为对输入查询的结果响应。执行器105表示运行时处理的执行层,该运行时处理基于提供给它们的增强的查询计划来处理查询。它们可以包含对数据元组流进行操作并以具体方式对数据元组流进行变换的运算符/操作实施方式(例如,SQL、图、空间等)。

[0100] 增强器103可以使用用于从由优化器114生成的查询计划中生成增强的查询计划的各种技术。在某些实施例中,增强器103可以使用由DAS102存储的元数据/语义信息117来进一步优化和增强由优化器114生成的查询计划。例如,基于可用于星形架构的语义信息(例如,来自星形架构语义模型),增强器103可以被配置为推断在由优化器114生成的查询计划中在事实记录的源的扫描操作期间适用的维度上下文/谓词条件。增强器103可以使用这个推断的信息来进一步优化由优化器114生成的查询计划。在某些实施例中,增强器103执行如下方法:利用在语义数据中定义的字段/列功能依赖性(例如,在星形架构或立方体的语义模型中)将所推断的维度上下文/谓词转换为对事实记录的源上的扫描操作的适用谓词条件。在某些情况下,增强器103可以将所推断的维度上下文/谓词条件应用于充分利用索引内的数据结构的事实记录的索引源(例如,OLAP索引事实源)。

[0101] 一般而言,出于本申请的目的,术语“事实源”用于指事实记录的源。事实源可以是作为事实(例如,事实记录)的源的任何关系。事实源可以是原始事实表或事实的某种其它物化。常见的物化技术包括预聚合的视图和立方体表示,诸如OLAP索引。例如,事实源可以是事实表(例如,基于星形架构的关系数据库中的事实表)、索引(例如,星形架构或立方体上的OLAP索引,诸如提供快速访问多维空间的任意子区域的事实的能力的预联接索引),或物化的视图(例如,对于星形架构在某个聚合级别的物化的视图)。事实源表示作为事实记录(例如,商业事实)的源的任何数据集或表。事实可以是在特定维度粒度。

[0102] 如上面所指示的,增强器103可以使用元数据/语义信息117来推断维度上下文/谓词条件。语义信息117可以包括商业情报信息。例如,语义信息117可以包括具有丰富价值的商业数据和基于现实世界的处理、组织结构、立方体信息等的结构语义。语义信息可以包括与商业模型有关的信息,如商业层次结构和数据中的功能依赖性。增强器103使用这个层次结构和依赖性信息来生成增强的查询计划,从而加快了查询的执行。例如,依赖性元数据信息可以包括描述数据库数据的不同字段(例如,列)之间的关系(例如,层次或非层次)的信息。例如,依赖性元数据可以包括描述“city”字段列的值和“state”字段列的值之间的层次关系的信息(例如,旧金山是加利福尼亚州内的城市)。增强器103可以使用依赖性元数据以及查询子计划结果的粒度来推断对查询子计划内事实源扫描的谓词条件适用性。在某些实施例中,可以在没有关于任何用户输入的情况下通过分析关于存储正由查询分析的数据的数据库的结构和架构的信息来确定语义信息。

[0103] 增强器103用于增强查询计划的语义信息可以从多个源获得或确定。作为一个示例,语义信息可以从商业智能(BI)数据源获得。例如,如图1B中所描绘的实施例中所示,DAS102可以包括或可以访问存储BI信息的各种数据源,其中BI信息包括语义信息117。在其它实施例中,语义信息可以是正被分析的数据的结构以及用于存储数据的数据结构。例如,对于存储正被分析的数据的数据库,该数据库的架构信息116可以被用于确定语义信息。增强器103可以使用架构信息116来推断谓词条件应用,并基于输入查询中的谓词进一步推断谓词条件。

[0104] 虽然在图1B中将优化器114和增强器103示为分开的子系统,但这并不是要进行限制。在替代实施例中,增强器103可以是优化器114的一部分。在这种实施例中,优化器114选择查询计划,然后使用本公开中描述的技术对其进行增强。在某些实施例中,增强器103可以是DAS102的一部分而不是DBMS104的一部分。在这种情况下,DBMS104可以与DAS102合作以生成增强的查询计划。在某些实施例中,解析器120、优化器114、增强器103和执行器105可以被统称为DBMS104的查询引擎。

[0105] 如上面所指示的,然后可以使用诸如SQL查询之类的分析查询来分析存储诸如图2中所描绘的示例的数据记录的数据库。典型查询可以涉及一个或多个事实源,这些事实源与维度表组合并彼此组合。在某些实施例中,如本文所公开的,增强器103从每个事实源扫描的角度分析由优化器生成的用于查询的查询计划,并尝试推断可以应用于事实行的维度上下文(例如,维度过滤器),以便被处理的事实行的数量大大减少和/或执行扫描操作的总成本降低。在星形联接(表示事实源与其维度上下文的组合的联接)的情况下,可以使用以下内容识别要推断并应用于事实源的各个维度上下文:

[0106] ●沿着星形架构的边缘联接的事实表示可以利用无损分解的组合(因此不会引入新的事实)。

[0107] ●可以利用星形架构的语义模型中的功能依赖性。

[0108] ●在OLAP索引事实源的情况下,可以执行重写。

[0109] 由于事实源可以将事实与其维度预联接,因此星形架构中的联接在事实源的上下文中被分类为事实源联接。星形架构联接是在形成事实源时要遍历的联接。

[0110] 可以接收要针对数据库数据200执行的分析查询,以分析存储在数据库200中的数据。这种查询的示例在下面被示为查询A:

[0111] 查询A

```
1) WITH customer_total_return AS
2) (    SELECT
3)      customer_key AS ctr_customer_key,
4)      store_key AS ctr_store_key,
5)      SUM(return_amt) AS ctr_total_return
6)      FROM store_returns sr，date d
7)      WHERE sr.date_key = d.date_key AND year = 2000
8)      GROUP BY customer_key，store_key)
9) SELECT customer_id
10) FROM customer_total_return ctr1，store s，customer c
11) WHERE ctr1.ctr_total_return >
12)     (SELECT AVG(ctr_total_return) * 1.2
13)     FROM customer_total_return ctr2
14)     WHERE ctr1.ctr_store_key = ctr2.ctr_store_key)
15)     AND store_key = ctr1.ctr_store_key
16)     AND state = 'TN'
17)     AND ctr1.ctr_customer_key = customer_key
18) ORDER BY customer_id
19) LIMIT 100
```

[0113]

[0114] 查询A的自然语言等同物是“识别‘有问题的’客户-商店关系,其中‘有问题的’被定义为那些退货额超过该商店的平均水平20％的那些客户”。问题是关于客户和他们所访问的商店的关系,而不是关于整体客户行为。另外,正在对Tennessee州进行分析。

[0115] 在接收到诸如上面所示的查询A之类的查询后,数据库管理系统被配置为解析查询,然后为该查询生成查询计划。然后由数据库管理系统针对存储的数据记录执行查询计划,以获得与查询有关的结果。图3是可以为查询A生成的查询计划300的逻辑树或图表示。查询计划300可以表示由优化器114生成和选择的查询计划。查询计划300表示如何对关系表中的数据进行操作的逻辑流程图。如前面所描述的,查询计划表示从查询计划管线中的一个操作到另一个操作的行源的流。查询计划的每个操作都从数据库中检索行,或从一个或多个其它操作(或行源)接受行。

[0116] 查询计划300包括表示事实或维度表的四个叶子节点。查询计划中的非叶子节点表示各种关系运算符。图3中所示的示例包括过滤器运算符、联接运算符、聚合运算符和投影运算符。非叶子节点将其子节点的行输出作为输入,并将该非叶子节点的行输出作为输

入提供给该非叶子节点的父节点,或者如果非叶子节点是根节点,那么根节点输出的行或记录作为查询执行的结果提供。例如,在图3中,节点310将节点308和304输出的行作为输入,节点308将节点306输出的行作为输入,节点304将存储表的扫描返回的行或记录作为输入。

[0117] 在查询计划300中,左聚合子计划302涉及计算Customer和Store粒度处的退货金额,而右聚合子计划312计算Store粒度的退货金额。然后将它们联接以提取有关Customer、Store组合的信息,这些组合的退货额超过相应的商店平均水平20%。此后,将在商店的州是Tennessee上对这些行进行过滤。

[0118] 查询一般而言涉及多层,其中一些是构建块层,其根据原始事实计算值。例如,可以将分析查询视为通过诸如联接、并集、排序和获取集合的前/后n个之类的操作进行组合的多维计算的层。甚至查询A提出的相对简单的问题都涉及对商店退货星形架构执行多次聚合-联接,如查询计划300所示。每个聚合-联接都涉及扫描非常大的事实表并与一个或多个维度表联接,其中大多数维度表相对于事实表较小。

[0119] 图3中描绘的查询计划300包括子计划302,其与查询A的第2-8行中的子查询对应。虽然在图3的示例中,子计划302与子查询对应,但是应当认识到的是,查询子计划不限于子查询。更确切地说,查询子计划可以与数据库查询的任何部分对应。

[0120] 子计划302包括用于评估查询A的第7行中的谓词的过滤操作和联接操作。根据计划300,将针对“日期”(“Date”)维度记录评估谓词条件“year=2000”,从而过滤它们。此后,谓词条件“sr.date_key=d.date_key”将通过将“Store_Returns”事实记录与过滤后的“Date”维度记录结合在一起进行评估。但是,执行联接操作可能需要大量计算,因为它通常涉及处理大量(例如,数百万个)事实记录。

[0121] 在未连接到子计划302的查询计划300的分开分支中,计划300还包括与查询A的第16行对应的过滤操作304(在state=“TN”上过滤)。根据计划300,过滤操作304将与子计划302的操作分开执行。此后,在经过节点306和308之后,经过滤的“Store”维度记录将与执行子计划302中的操作的一些结果联接。

[0122] 计划300涉及对事实表Store_Returns的两次事实扫描。但是,整个查询的上下文是Tennessee州和2000年。但是,在计划300中,包括对事实源Store_Returns进行扫描的子计划302并不仅将记录限制为州为Tennessee州的记录。因此,还提供了非Tennessee州记录,用于处理子计划302中的联接操作以及从子计划302到联接操作306。因此,这些事实记录(即所有州在2000年的商店退货)将不必要地被结转到操作306、308和310,因为在310处,不在TN中的那些事实记录将不会被联接并被过滤掉。携带这些不必要的记录通过306、308和310在执行查询计划300所需的资源方面是浪费的,因为在查询计划中结转分记录的数量,并且增加了执行查询计划300所需的时间(即执行查询需要花费更多时间)。这导致大量的计算资源浪费,并导致执行时间增加。

[0123] 在某些实施例中,增强器103被配置为从查询计划的一部分推断TN过滤条件并改变查询计划,使得推断出的条件被应用于查询计划的另一个无关部分,从而减少正被携带通过整个查询计划的执行的事实记录的数量和/或减少执行扫描操作的总成本。重写查询计划有助于避免不必要地处理和结转事实记录。可以使用从星形架构信息116和/或语义元数据117得出的推断来重写查询计划300。增强器103可以使用星形架构116和/或语义元数

据117来重写查询计划300。

[0124] 例如,如图4中所描绘的,星形架构400指示“Store”维度记录可以与“Store_Returns”事实记录联接。因此,增强器103可以推断出用于过滤“Store”维度记录的谓词条件也可以用于过滤“Store_Returns”事实记录。

[0125] 基于从星形架构116和/或语义元数据117得出的推断,增强器103可以通过将查询计划中的谓词条件从查询计划中的查询子计划外部传播到查询子计划内部来重写和增强查询计划300。未连接到查询计划的子计划或子部分的谓词条件被插入到该子计划或子部分中或与该子计划或子部分相关联,以使得在重写之后从子计划输出的事实记录的数量少于在重写之前从子计划输出的事实记录的数量,和/或减少在子计划中执行扫描操作的总成本,而不会影响从查询执行获得的总体查询结果。以这种方式,作为重写的结果,作为重写的结果,将在原始查询计划的较后期操作期间被过滤掉的不相关的事实记录现在将在查询计划的较早期被过滤掉,从而减少在执行查询计划时必须结转的事实记录的数量并避免不必要的后续处理。如上面所指示的,事实记录可能以百万计。因此,减少在查询计划管线中从查询计划中的一个操作到查询计划中的另一个操作必须被结转的事实记录的数量显著改善了整体查询执行(即,使执行更快)并减少可观数量的计算资源(例如,I/O资源、处理(例如,CPU)资源和存储器资源)的浪费。

[0126] 图5是根据某些实施例的可由增强器103生成的、用于执行查询A的重写的增强的查询计划500的逻辑树或图表示。计划500表示图3中描绘的计划300的重写版本。计划500包括子计划502,它与查询A的第2-8行中的子查询对应。在图5中,图3中的子计划302已被重写为子计划502。与子计划302相比,重写后的子计划502不仅具有“year＝2000”条件,而且现在还具有附加条件“state＝TN”)。以这种方式,来自查询计划300中的操作304的“state＝TN”谓词条件已经作为附加条件(除了“year＝2000”之外)被传播到子计划502中的过滤操作。具有这两个条件的这个重写的过滤操作504被应用于从“Store_Returns”事实表扫描的事实记录。因此,通过将过滤条件从图3中的过滤操作304传播到子计划502中的过滤操作504,子计划302(或更具体而言,过滤操作304)已经被增强器103重写为子计划502。

[0127] 以类似的方式,在图5中,图3中的子计划312已经被重写为子计划512。与子计划312相比,重写的子计划512不仅具有“year＝2000”条件,而且现在还具有附加条件“state＝TN”)。以这种方式,来自查询计划300中的操作304的“state＝TN”谓词条件已经作为附加条件(除了“year＝2000”之外)被传播到子计划512中的过滤操作。具有这两个条件的这个重写的过滤操作506被应用于从“Store_Returns”事实表扫描的事实记录。因此,通过将过滤条件从图3中的过滤操作304传播到子计划512中的过滤操作506,增强器103已经将子计划312(或更具体而言,过滤操作304)重写为子计划512。

[0128] 以上述方式重写查询计划使得能够从事实源Store_Return过滤掉与Tennessee州无关的任何事实记录,从而在进一步处理这种不相关的事实记录时不会浪费计算资源。与子计划302相比,由过滤操作504和子计划502输出并可供项目计划500中的进一步操作(例如,联接操作508)使用的事实记录的数量比由子计划302(具体而言,由子计划302中的联接操作)输出并可供查询计划300中的下一个操作使用的记录的数量少得多。这是因为过滤操作504和子计划502仅将与TN条件匹配的记录输出到计划500中的下游操作(例如,获得由子计划502输出的事实记录的联接操作);非Tennessee州的记录被过滤掉了。以类似的方式,

与子计划312相比,由过滤操作506和子计划512输出并可供项目计划500中的进一步操作(例如,联接操作508)使用的事实记录的数量比由子计划312(具体而言,由子计划312中的联接操作)输出并可供查询计划300中的下一个操作使用的记录的数量少得多。这是因为过滤操作506和子计划512仅将与TN条件匹配的记录输出到计划500中的下游操作(例如,获得由子计划512输出的事实记录的联接操作508);非Tennessee州的记录被过滤掉了。这可以大大减少提供给联接操作的记录的数量;取决于事实表的尺寸,事实记录的数量的减少等于或大于100倍。由于子计划502和512输出的事实记录的数量的这种减少,计划500以及因此为其生成查询计划500的相应查询执行得更快(由于执行所需的CPU周期更少)并且在许多情况下在分析相同数据并返回相同查询结果的同时使用的存储器资源将比未增强的查询计划300少。

[0129] 在图3和图5中所描绘的示例中,来自查询计划的一部分的条件被传播到查询计划的另一个部分。在这个示例中,谓词条件(基于State＝TN的过滤)从查询计划的一部分(例如,从图3中的过滤操作304)传播到查询计划的另一个部分(例如,到图5中的过滤操作504和506)。在这个示例中,谓词条件被传播到的查询计划图的操作或部分既不是谓词条件从中传播的树的那部分的后代也不是其祖先,即,在查询计划300或500中过滤操作504和506不是过滤操作304的后代或祖先。

[0130] 虽然在图5的示例中来自过滤操作304的谓词条件被传播到子计划302和312以创建重写的子计划502和512,但是应当认识到的是,这仅仅是示例。一般而言,来自查询计划图中的第一操作的谓词条件可以被传播并与查询计划中的不同部分或操作相关联,以生成优化的增强的重写的查询计划。这可以取决于可用的数据结构(例如,物化的视图)。

[0131] 在某些实施例中,作为分析查询计划300的一部分,增强器103集中在聚合-联接子计划上,这些子计划是整个查询计划中仅包含聚合、联接、投影、过滤或扫描运算符/操作的子计划。这些的示例是图3中描绘的子计划302和312。增强器103的目标是找到可推到这些子计划内的对事实源关系进行扫描的谓词条件或过滤器。

[0132] 在某些实施例中,增强器103寻找的通用模式(模式A):

[0133] -查询计划中表示联接操作的节点(“联接节点”),该节点定义事实源扫描(fs)与某个其它表扫描(ot)之间的联接条件。联接条件可以被表示为事实源的列与另一个表之间的相等性检查的列表(fs.col1＝ot.col1和fs.col2＝ot.col2和…)。

[0134] -事实源侧不应当为空供应,即,沿着从事实源扫描到所考虑的联接(j1)的路径,不得存在事实源侧为空供应的另一个联接(j2)(即,fs不能是j2的子c1的后代,其中j2是外联接,而c1是j2中的空供应侧)。

[0135] 模式A如下所示。模式A第2行和第4行替代聚合/联接/投影/过滤操作的任意组合。

[0136] 1.联接(fs.col1＝ot.col1和fs.col2＝ot.col2和…)

[0137] 2.投影/过滤/聚合操作

[0138] 3.事实源扫描(fs)

[0139] 4.投影/过滤/聚合操作

[0140] 5.其它表扫描(ot)

[0141] 如果在整体查询计划中找到与上述模式A匹配的子计划,那么可以对事实源(模式A中的fs.col1、fs.col2、...)的联接列应用“半联接”限制。事实源行可以限制为其连接列

值与OtherTable中的某行(在ot.col1、ot.col2...上)匹配的行。另外,如果OtherTable具有某个有效的过滤器,那么事实源关系扫描可以被重写,如下所示:

[0142] 1.左半联接(fs.col1＝ot.col1和fs.col2＝ot.col2和…)

[0143] 2.事实源扫描(fs)

[0144] 3.选择不同的ot.col1、ot.col2、…

[0145] 4.过滤

[0146] 5.OtherTable扫描(ot)

[0147] 这个重写的潜在优势是减少了在查询计划管线的后续运算符中处理的事实源行。实际上,源行的减少非正式地取决于OtherTable上过滤器的选择性。过滤器的选择性越高(例如,具有大量不同值的列上的相等条件),事实源行中减少的机会就越大。在星形架构联接的情况下,联接将进行无损分解(联接不会添加新事实),因此维度过滤器对所扫描事实的好处很有可能与过滤器的选择性成正比。

[0148] 这个重写的成本基于半联接操作的执行方式。在一般情况下,这是使用左半联接(Left Semi-Join)运算符完成的,该运算符通常将涉及联接物理计划。

[0149] 有多种技术可用于在查询计划内高效地传播谓词条件。可以基于各种因素来选择所使用的特定技术,这些因素包括但不限于估计的计算节省和/或某些数据结构的可用性。

[0150] 一种技术涉及在事实记录和维度记录之间执行半联接操作。这可以基于针对维度记录评估谓词条件以确定满足该谓词条件的维度键集合来实现。这个维度键集合在本文中被称为“键入列表”(“key in-list”)。

[0151] 图6描绘了用于生成可以用于高效地评估查询A的键入列表602和604的示例处理600。更具体而言,针对维度表204评估谓词条件“year＝2000”以确定键入列表602,并针对维度表208评估谓词条件“state＝'TN'”以确定键入列表604。键入列表是维度表中的维度记录中满足谓词条件的外键的列表。例如,对于Date维度表204,图6中所描绘的第一条记录满足“year＝2000”并且date_key＝1是来自那个记录的外键。键入列表602和604可以用于扫描和过滤事实记录,从而将谓词条件传播到事实记录。

[0152] 但是,执行半联接操作会招致大量的计算开销。扫描事实记录可以是计算密集的,因为通常会有大量事实记录。此外,生成键入列表增加了执行半联接操作的计算开销。值得一提的是,减少原本会不必要处理的事实记录(例如,与Tennessee州无关的事实记录)应当超过这个开销。因此,增强器103可以在重写查询计划以包括半联接操作之前估计计算节省。

[0153] 在多个谓词条件的情况下,查询增强器103可以分析相继执行半联接操作的增量节省。这可以涉及将谓词条件放置在由于单独应用每个谓词条件而产生的计算节省的降序中。然后,增强器103可以按这个次序分析谓词条件,以估计来自相继应用谓词条件的增量节省。因此,可以指定半联接操作以应用相继谓词条件,只要增量节省不是无关紧要的即可。

[0154] 用于将谓词条件高效地传播到事实记录的另一种技术涉及与维度值联接的事实值(例如,事实度量值或来自事实表的任何其它值)的表格表示。还涉及使得能够高效访问物化的视图中包括的维度值的存储位置的一个或多个索引(例如,倒排索引)。总的来说,具有一个或多个索引的表格表示在本文中被称为“OLAP索引”。通常,OLAP索引是单个复合数

据布局格式,其被优化以基于谓词条件扫描行的任意子集,前提是使用诸如扫描优化的列编码、存储器优化的列布局、向量化的扫描操作之类的技术;谓词评估也得到了高度优化,例如作为位集操作的集合。这些特征使得OLAP索引理想地适于自组织分析工作负载。

[0155] 图7描绘了包括维度索引704和706的示例OLAP索引700。OLAP索引700还包括表702,该表702包括通过将事实表202与维度表204和208联接而得到的数据。如图7中所描绘的,OLAP索引包括列状格式的数据以及维度值上的一个或多个倒排索引。对于每个维度值,维护位置位图。

[0156] 在许多数据库中,字段(例如,列708或710)中的值连续存储在一定范围的存储器地址内。例如,当从存储装置将数据读取到系统存储器中时,列710的值可以连续地存储在易失性存储器中,作为可以被高效访问的列状数据库数据。为了提高访问特定列值的效率,索引(例如,维度索引)可以存储字段值和存储器位置之间的映射。在图7的示例中,维度索引704指示列710的“state”维度值712的存储器位置714,并且维度索引706指示列708的“year”维度值716的存储器位置718。

[0157] 因此,OLAP索引使得能够节省计算资源。由于OLAP索引包括维度值,因此可以直接针对OLAP索引评估谓词条件,而无需生成入列表(in-list)。此外,维度值位图表示从维度值到具有该维度值的事实行的子集的映射;每个维度谓词条件可以被映射到位图或位图组合:例如State=’TN’是’TN’值位图中的事实行,State>’TN’是位图中通过OR组合>’TN’的所有State值的所有位图而获得的事实行;多个维度谓词条件的进一步评估也可以被评估为非常快速的位图组合函数,如AND、OR、NOT,而不必扫描实际事实数据。诸如图7中所描绘的之类的索引使得谓词条件(诸如’state=“Tennessee”’和’date=“2000”’)能够以快速方式计算。OLAP索引物化使得能够执行非常高效的“跳过扫描”。谓词条件(例如,’state=“Tennessee”’或’date=“2000”’)的评估生成行标识符的列表,从而识别具有与谓词条件匹配的数据的事实行。例如,行标识符1、4、5、7、9、10、13、15、17、19、20、23和24与满足’state=“Tennessee”’条件的事实记录对应。作为另一个示例,行标识符1、2、3、4、5、6、7和8与满足’date=“2000”’条件的事实记录对应。在事实扫描操作期间,可以使用rowid列表跳过大量事实记录行。通常,分析查询集中在广阔多维空间的一小个区域上,因此跳过扫描的执行速度明显快于扫描所有事实记录(或事实行)然后消除那些不满足过滤谓词条件的扫描。因而,将OLAP索引与谓词条件一起使用使得事实扫描操作更快。

[0158] 虽然为了清楚起见图7将存储器位置714和718描绘为十进制值,但是应当认识到的是,存储器位置714和718可以被表示为二进制值、十六进制值等。在一些实施例中,存储器位置714和718可以作为位置位图来维护,从而使得计算上便宜和快速的位图AND操作能够将多个谓词条件同时传播到事实记录。此外,虽然为了清楚起见图7将存储器位置714和718描绘为相对地址(例如,字节偏移量),但是应当认识到的是,在一些实施例中,存储器位置714和718可以被表示为绝对地址。

[0159] 一般而言,针对OLAP索引评估谓词条件的收益应当始终超过任何计算成本。因此,不需要增强器103执行成本-收益分析。但是,合适的OLAP索引可能并不总是可用。例如,可以不存在包括“state”维度值的OLAP索引,针对该维度值评估谓词条件“state=’TN’”。

[0160] 用于将谓词条件高效地传播到事实记录的另一种技术涉及上面提到的技术的混合,并且在本文中被称为“索引半联接”技术。顾名思义,索引半联接操作是涉及OLAP索引的

特定类型的半联接操作,虽然OLAP索引不适合直接评估谓词条件。

[0161] 图8A描绘了示例OLAP索引800,其包括表802和维度索引804。表802包括通过联接事实表Store_Returns 202与维度Store表208而得到的数据。列806的值可以作为列状数据库数据连续地存储在易失性存储器中,并且维度索引804可以被用于高效地访问特定列值。更具体而言,维度索引804将“city”维度值808映射到表内的存储器位置810。

[0162] 虽然为了清楚起见图8A将存储器位置810描绘为十进制值,但是应当认识到的是,存储器位置810可以被表示为二进制值、十六进制值等。在一些实施例中,可以将存储器位置810作为位置位图来维护,从而使计算上便宜且快速的位图AND操作能够将多个谓词条件同时传播到事实记录。此外,虽然为了清楚起见图8A将存储器位置810描绘为相对地址(例如,字节偏移量),但是应当认识到的是,在一些实施例中,存储器位置810可以被表示为绝对地址。

[0163] 但是,维度索引804不适合直接评估谓词条件“state＝'TN'”。这是因为维度索引804基于“city”值,而不是“state”值。

[0164] 不过,基于索引半联接操作,维度索引804仍可以被用于将谓词条件“state＝'TN'”传播到事实记录。这可以涉及针对维度记录评估谓词条件,以确定满足谓词条件并包括在维度索引804中的维度值(例如,非键值)集合。这个维度值集合在本文中称为“值入列表”(“value in-list”)。

[0165] 图8B描绘了用于生成值入列表822的示例处理820,该列表822可以被用于针对维度索引804评估谓词条件“state＝'TN'”。更具体而言,针对维度表208评估谓词条件“state＝'TN'”,以确定包括一个或多个“city”值的值入列表822。在图8B的示例中,这返回单个城市值“Nashville”。值入列表822(例如,“Nashville”)随后可以被用于扫描维度索引804以查找与表802中的相关记录对应的存储器位置(例如,Nashville的位置1、4、5、7、9、10、13、15、17、19、20、23以及24),从而在不扫描表802的所有记录的情况下过滤表802的记录。

[0166] 虽然执行索引半联接操作通过避免扫描所有事实记录来节省计算资源,但它会招致与生成值入列表相关联的计算开销。值得一提的是,减少原本将不必要地被处理的事实记录(例如,与Tennessee州无关的事实记录)应当超过这个开销。因此,增强器103可以在重写查询计划以包括索引半联接操作之前估计计算节省。

[0167] 在多个谓词条件的情况下,增强器103可以分析相继执行索引半联接操作的增量节省。这可以涉及将谓词条件放置在由于单独应用每个谓词条件而产生的计算节省的降序中。然后,增强器103可以按这个次序分析谓词条件,以估计来自相继应用谓词条件的增量节省。因此,可以指定半联接操作以应用相继谓词条件,只要增量节省不是无关紧要的即可。

[0168] 如上面所指示的,不仅可以从星形架构信息116中得出推断,而且还可以从语义元数据117中得出推断。从语义元数据117得出的推断可以被用于转化谓词条件,以便可以将其传播到事实记录。例如,语义元数据117可以包括指示以周、月、季度等表示的时间值之间的层次关系的信息。然后,这个信息可以被用于将诸如谓词条件“quarter＝2”之类的条件转化成“month BETWEEN 4AND 6”,使得可以针对“month”值评估谓词条件,因为图2中描绘的Date维度表204包括month、day和year字段,但不包含用于季度的字段。以这种方式,可以将基于不是维度表中的列的值的谓词条件转化成是维度表中的列的值。

[0169] 图9描绘了根据某些实施例的用于基于功能依赖性将谓词条件传播到事实记录的示例900。在图9的示例中,“quarter＝2”是未转化的谓词条件904,可以识别该条件以传播到图9中所描绘的OLAP索引909,该OLAP索引909包括表908和维度索引910。但是,表908中的列912、914、916或918都不是“quarter”列,这使得难以对照OLAP索引910评估谓词条件904。

[0170] 不过,语义元数据901可以包括依赖性元数据,其可以被用于基于索引910中的维度属性将谓词条件904转化成谓词条件906。更具体而言,语义元数据901可以包括日期层次902(例如,月份到季度、季度到年),其可以用于将谓词条件“quarter＝2”转换成基于月的谓词条件“month BETWEEN 4AND 6”。此后,可以对照维度索引910评估经转化的谓词条件906,该维度索引910将月份值920映射到存储位置922。在这个示例中,4、5和6月满足谓词条件,并映射到它们的存储器位置(或rowid)。虽然图9描绘了涉及OLAP索引的示例处理900,但是应当认识到的是,可以使用本文所述的任何适当技术将经转化的谓词条件传播到事实记录的任何源。

[0171] 在图9中所描绘并且在上面描述的示例中,将基于季度的谓词条件替换为基于月份的约束谓词条件。替换约束谓词条件不必等同于原始谓词条件;它只应当返回由它所替换的谓词条件返回的所有行(事实记录)。关于等式和约束条件(＝、!＝、<、>、<＝、>＝、in、between)的替换策略用于基于级别的层次结构。对于合理尺寸的层次结构,可以在增强的查询计划生成期间由增强器103维护并使用层次结构的内存(in-memory)表示,以供计算替换谓词条件。

[0172] 由于目标是减少通过执行查询计划而处理的事实记录的数量和/或减少在查询计划中执行扫描操作的总成本,因此谓词条件的应用在实现这个目标中起着重要的作用。在图9中所描绘并且在上面描述的示例中,查询中提供的谓词条件(“quarter＝2”)与替换所提供的谓词条件的谓词条件(“month BETWEEN 4AND 6”)不同。即使两个谓词条件不同,替换谓词条件的应用也确保替换谓词条件也返回由所提供的谓词条件返回的任何行。在查询计划的性能和查询的性能方面,这提供了巨大的好处。可以通过遍历功能依赖性来推断替换谓词条件。因此,在季度->月->周日期层次结构中,如果我们具有关于周的谓词条件,那么如果week列不可推到事实源,而事实源可以处理季度谓词条件,那么我们可以推入对应的季度谓词条件,例如:可以将week＝1推为quarter＝1,可以将week>25推为quarter>2,依此类推。谓词条件转化基于遍历定义列之间的依赖性的关系树。如果将这些树维持为内存结构,那么转化非常快并且不会为计划增加太多开销。层次结构肯定是这种情况,在这种情况下,通常支持通过内存数据结构进行成员导航。

[0173] 虽然图9描绘了涉及在未转化的谓词条件904中指定的字段与在经转化的谓词条件906中指定的字段之间的层次关系的示例,但是应当认识到的是,本文所述的技术不限于表现出层次关系的字段。例如,语义元数据901可以包括可以用于执行谓词条件的转化的其它类型的信息。例如,语义信息可以包括指示在南方的商店不出售冬季服装的信息,从而使得能够将以冬季服装表达的谓词条件转化成以地理区域表达的谓词条件。

[0174] 在本公开中描述的任何技术可以被用于将谓词条件传播到事实记录。谓词条件的传播表现出传递特性,因为谓词条件可以从维度记录传播到事实记录,而这些维度记录可能不会直接与事实记录联接。例如,谓词条件可以跨多个维度表或跨另一个事实表传播。

[0175] 在一些实施例中,谓词条件可以从维度记录的第一集合跨维度记录的第二集合被

传播到事实记录的集合。可以参考图10A中描绘的示例来说明。图10A描绘了示例数据库数据1000,其包括“Store_Returns”事实表202、“Customer”维度表1002和“Customer_Address”维度表1004的缩略形式。可以基于列212和1006将“Store_Returns”事实表202直接与“Customer”维度表1002联接,并且可以基于列1008和1010将“Customer”维度表1002直接与“Customer_Address”维度表1004联接。但是,在图10A的示例中,没有可以用于直接联接“Store_Returns”事实表202与“Customer_Address”维度表1004的列。

[0176] 不过,增强器103可以使用星形架构信息116来推断谓词条件“state＝'TN'”可以从“Customer_Address”维度表1004传播到“Store_Returns”事实表202,因为它们各自与“Customer”维度表1002相关。基于这个推断,增强器103可以检查将“Store_Returns”事实表202的列与“Customer_Address”维度表1004的列相关的表(或事实记录的任何其它源)。

[0177] 图10B仅描绘了这样的表1020。在图10B的示例中,表1020包括由联接操作产生的数据,该联接操作涉及使用列212和1006的“Store_Returns”事实表202和“Customer_Address”维度表1004,并且在“Customer”维度表1002和“Customer_Address”维度表1004之间使用列1008和1010。因此,表1020包括与“Customer_Address”维度表1004的“state”列1012对应的“state”列1022。

[0178] 本文描述的任何技术可以被用于将谓词条件“state＝'TN'”传播到表1020。例如,如果对于表1020中的列1022的值存在倒排索引,那么可以对照该倒排索引来评估谓词条件以过滤表1020,以仅扫描满足“state＝'TN'”谓词条件的事实行。

[0179] 对于图10A和图10B中所描绘的示例,如果满足以下条件,那么可以传播谓词条件:

[0180] • 已经推断出事实源Store_Returns表202(FS1)可以基于维度表Customer 1002(T1)的过滤而减少。

[0181] • FS1和T1之间的联接是星形架构。这意味着这种联接操作不会向联接添加或移除任何输入事实记录。

[0182] • T1和维度表Customer_Address 1004(T2)之间的联接

[0183] -在T1侧不是空供应

[0184] -在FS1.StarSchema中是星形架构联接然后:如果对于联接T1-T2中的T1中的每个T1联接列jc,在FS1中存在等效列$ec_i$,那么可以对FS1中的$ec_i$列应用semiJoin限制。

[0185] 在一些实施例中,谓词条件的评估可以从维度记录的集合(来自维度表DT)跨事实记录的第一集合(事实源1“FS1”)被传播到事实记录的第二集合(事实源2“FS2”)。例如,当星形架构包括多个事实记录集合时,这可以是可能的。图11描绘了这种架构1100的示例。架构1100包括两个事实表(示为具有被强调的边界):

[0186] “Store_Returns”事实记录表1102和“Store_Sales”事实记录表1104。其它表(Date、Store、Item、Customer、Customer_Address)是存储用于事实记录的上下文信息的维度表。这两个事实表每个均能够与这些维度表(诸如与“Store”维度记录表1106)联接。

[0187] 图12A描绘了可以在架构1100中描述的示例数据库数据1200。数据库数据1200包括“Store_Returns”事实表1202、“Store”维度表1204和“Store_Sales”事实表1206。虽然图12将数据库数据1200描绘为表,但是应当认识到的是,数据库数据1200可以被表示为索引、物化的视图和/或任何其它元组集。

[0188] “Store_Returns”事实表1202和“Store”维度表1204可以基于列1208和1210联接,

这两个列表现出外键-主键关系。因此,涉及列1212和1214中的任何一个的谓词条件(例如,“state＝’TN’”)可以使用在此描述的任何技术从“Store”维度表1204传播到“Store_Returns”事实表1202。

[0189] 但是,“Store_Sales”事实表1206也可以与“Store”维度表1204联接。这是因为列1216和1210也表现出外键-主键关系。因此,基于图11中描绘的架构1100,增强器103可以推断出也可以使用本文描述的任何技术将“Store_Returns”事实表1202的任何过滤应用于“Store_Sales”事实表1206。

[0190] 图12B描绘了包括查询子计划1222、1224和1226的示例逻辑查询计划1220。根据查询计划1220,在执行子计划1222中的操作的结果与执行子计划1224中的操作的结果之间执行联接操作1230。此外,计划1220在执行联接操作1230的结果与执行子计划1226中的操作的结果之间指定执行联接操作1232。

[0191] 虽然过滤操作1228包括在子计划1222中,但是过滤操作1228不包括在子计划1224和1226中。当执行联接操作1230时,这会导致来自“Store_Returns”事实表1202的所有事实记录的大量不必要的处理。以类似的方式,当对从子计划1226从“Store_Sales”事实表1206返回的所有事实记录执行联接操作1232时,会执行大量不必要的处理。这种不必要的处理会大大增加执行查询计划1220所花费的总时间。

[0192] 图12C描绘了示例重写的增强的查询计划1240,其已基于查询计划1220进行了重写,以减少查询计划所处理的事实记录的数量。根据计划1240,过滤操作1228中的谓词条件“State＝TN”不仅传播到子计划1224中作为过滤操作1242,而且还传播到子计划1226中作为过滤操作1244。因此,由子计划1224输出并提供给联接操作1230的事实记录的数量(假设Store_Returns表保持州不是TN的记录)少于查询计划1220中提供给联接操作1230的事实记录的数量。这是因为,在计划1240中,只有来自事实表Store_Returns的满足“state＝’TN’”谓词条件的事实记录才提供给联接操作1230,而不是计划1220中的所有事实记录。以类似的方式,由子计划1226输出并提供给联接操作1232的事实记录的数量(假设Store_Sales表保持州不是TN的记录)少于查询计划1220中提供给联接操作1232的事实记录的数量。这是因为,在计划1240中,只有来自事实表Store_Sales的满足“State＝’TN’”条件的事实记录才被提供给联接操作1232,而不是计划1220中的所有事实记录。因此,通过首先过滤掉不满足维度谓词条件“State＝’TN’”的“Store_Returns”事实表1202和“Store_Sales”事实表1206的任何行,可以显著减少执行联接操作1230和1232的计算开销。因此,查询计划1240比未增强的查询计划1220执行得快(例如,使用更少的CPU周期),并且在许多情况下可以使用更少的存储器资源。

[0193] 如上所述,在某些实施例中,公开了用于在涉及相同事实(涉及对事实的多个粒度的分析的查询)或不同事实(例如,涉及比较/组合不同活动的查询)的子查询中传递式传播上下文的技术)。在某些实施例中,如果以下条件成立,那么可以应用这些技术:

[0194] • 已经确定基于维度表DT在事实源FS2上的维度上下文应用。

[0195] • 在DT的键列上存在另一个事实源FS1与FS2的联接,并且DT也是FS2星形架构中的维度表。

[0196] 如果上述条件成立,那么可以将来自DT的维度上下文应用于FS1。

[0197] 图13A是描绘根据某些实施例的用于生成增强的查询计划的方法的简化流程图

1300。处理1300可以在由相应系统、硬件或其组合的一个或多个处理单元(例如,处理器、核)执行的软件(例如,代码、指令、程序)中实现。软件可以存储在非暂态存储介质上(例如,在存储器设备上)。图13中呈现并在下面描述的方法旨在是说明性而非限制性的。虽然图13描绘了以特定顺序或次序发生的各种处理步骤,但这并不意味着是限制性的。在某些替代实施例中,这些步骤可以以某种不同的次序执行,或者一些步骤也可以并行执行。

[0198] 在1302处,可以接收第一或原始查询计划。原始查询计划可以是为被提供给图1A和图1B中描绘的分析平台100的数据库查询(例如,SQL查询)而生成的计划。该查询可以是针对存储在数据库中的非预聚合的数据运行的自组织查询。参考图1B中描绘的实施例,原始查询计划可以由优化器114生成。

[0199] 原始查询计划可以包括操作(或运算符)的管线。操作(运算符)可以包括例如事实扫描(有时简称为扫描操作)、过滤、联接、聚合、投影和其它操作。原始查询计划中的操作可以在管线中进行分层组织,使得将操作的输出作为输入提供给其它(一个或多个)下游操作。可以将查询计划分层表示为包括父节点和子节点的带根的树。在这种表示中,从由树的叶子节点表示的操作开始执行操作。然后将来自叶子节点的输出提供给父节点,将来自父节点的输出提供给这些父节点的父节点,依此类推,直到提供来自根节点的输出的根节点是查询(为该查询而生成查询计划)的结果为止。例如,在1302中接收的原始查询计划可以如图3和/或图12B中所描绘的。

[0200] 原始查询计划中的一个或多个操作可以被分组以形成子计划。例如,原始查询计划可以包括第一子计划,该第一子计划包括事实表扫描操作,该操作涉及扫描事实记录的源(例如,事实表)。第一子计划还可以包括其它零个或多个其它操作,诸如过滤、联接、聚合、投影操作。例如,图3中所描绘的子计划302以及图12B中所描绘的子计划1222、1224和1226。在原始查询计划中,把由第一子计划输出(即,从第一子计划中输出作为在第一子计划中执行操作的结果)的事实记录作为输入提供给原始查询计划管线中的父操作。

[0201] 在1304处,通过重写原始查询计划来生成增强的查询计划。作为重写的结果,在所得的增强的查询计划中,至少一个维度谓词条件的评估被传播到增强的查询计划中的事实扫描操作并与该事实扫描操作相关联,其中该谓词条件不与原始查询计划中的相同事实扫描操作相关联。例如,可以将谓词条件插入第一子计划中并与第一子计划中的至少一个事实扫描操作相关联。在增强的查询计划中,基于将该谓词条件传播到事实表,作为子计划的一部分对谓词条件进行评估。作为谓词条件与第一子计划和事实扫描操作的传播和关联的结果,增强的查询计划比原始查询计划的执行得更快。例如,在一些情况下,由于将该谓词条件与该事实扫描操作相关联,与原始查询计划相比,由增强的查询计划处理的事实行/记录更少。这是因为仅满足谓词条件的那些事实记录在查询计划管线中被提供给父操作或下游操作;不满足谓词条件的事实记录将被过滤掉,并且不提供给父操作,该父操作可以是例如联接操作。这导致提供给查询计划管线中的下一个和后续操作的事实记录数量减少。这减少了与随后执行查询计划管线中的联接操作和其它后续操作相关联的计算开销。由于需要在查询计划管线中处理和结转较少的事实记录,因此增强的查询计划比原始查询计划执行得更快并且使用更少的资源(例如,处理资源、存储器资源)。

[0202] 在一些其它情况下,与事实扫描操作相关联的谓词条件是,即使可能不会减少从事实扫描操作输出到查询计划管线中的下游操作的事实记录的总数,但是与原始查询计划

中的相同事实扫描操作相比,与执该行事实扫描操作相关联的成本(例如,执行该事实扫描操作所需的CPU周期或时间)减少了。

[0203] 因此,在1304中重写的增强的查询比在1302中接收到的原始查询计划执行得更快。因此,为其生成增强的查询计划的查询执行得更快,并且可能使用更少的存储器资源。因此,增强的查询计划是对原始查询计划的增强和改进。在图1B所示的实施例中,1304中的处理可以由增强器103执行。

[0204] 在1306处,可以执行在1304中生成的增强的查询计划。如上面所指示,由于将一个或多个谓词条件传播到查询计划中的一个或多个操作,因此增强的查询计划比未增强的原始查询计划执行得更快(即,使用更少的时间),并且在许多情况下使用更少的存储器资源。

[0205] 在1308处,作为在1306中执行增强的查询计划的结果而从数据库检索到的事实记录可以作为对为其生成了原始查询计划和增强的查询计划的查询的响应而输出。在图1B中所描绘的实施例中,可以由执行器105执行1306和1308中的处理。

[0206] 图13B是描绘根据某些实施例的用于生成增强的查询计划的方法的简化流程图1320。图13B中的处理可以在由相应系统、硬件或其组合的一个或多个处理单元(例如,处理器、核)执行的软件(例如,代码、指令、程序)中实现。软件可以存储在非暂态存储介质上(例如,在存储器设备上)。图13B中呈现并在下面描述的方法旨在是说明性而非限制性的。虽然图13B描绘了以特定顺序或次序发生的各种处理步骤,但这并不意味着是限制性的。在某些替代实施例中,这些步骤可以以某种不同的次序执行,或者一些步骤也可以并行执行。

[0207] 在某些实施例中,可以将图13B中描绘的处理作为图13A中1304中执行的处理的一部分来执行。对于图1B中描绘的实施例,图13B中描绘的处理可以由增强器103执行。增强器103的输入包括针为查询生成的原始查询计划,例如,针对SQL查询生成的SQL查询计划。增强器103还可以访问元数据信息和架构信息(诸如关于关系架构、星形架构定义、表的列之间的功能依赖性的信息)、数据的可用物理结构(诸如星形架构上的OLAP索引)等。出于图13B和随附描述的目的,假设元数据和架构信息存储在元存储库中。图13B中描绘的处理的输出是增强的查询计划,其执行得比原始查询计划快,并且在许多情况下,使用的资源少于原始查询计划,同时返回与原始查询计划相同的结果。如上所述,增强的查询计划比原始查询计划执行得更快,因为增强的查询计划处理的事实记录的数量少于原始查询计划,和/或增强的查询计划中处理事实记录的成本比原始查询计划中的少,因为事实记录得到了更高效的处理。

[0208] 如图13B中所描绘的,在1322处,分析原始查询计划以识别原始查询计划中的事实表。这可以使用各种不同的技术来执行。例如,原始查询计划中的事实表可以通过将原始查询计划中的表与元存储库中的事实表进行匹配、通过查询级别或会话/连接级别的用户提示以及其它技术来识别。

[0209] 在1324处,对于在1322中识别出的每个事实表,在原始查询计划中对识别出的事实表上的每个事实扫描运算符执行查询计划遍历,以计算与这个事实扫描运算符的确认联接的集合并确定/推断任何适用的维度上下文谓词条件。在某些实施例中,在原始查询计划中对事实表扫描运算符进行遍历涉及从事实扫描运算符开始并遍历原始查询计划,直到到达原始查询计划的根节点。例如,对于图3中所描绘的查询计划,对子计划302中的Store_Returns事实表扫描操作的遍历涉及从这个节点320开始并访问其所有祖先节点:节点322、

节点324、节点306,以此类推,直到并包括根节点326。

[0210] 为了在遍历期间为事实表的事实扫描操作识别/推断适用的维度上下文谓词条件,增强器103维持每个被拜访的节点操作/运算符的输出的“粒度”。查询计划中任何关系运算符的输出都是计算出的数据集。任何表(基础数据集)或计算出的数据集的“粒度”是其架构中的维度列的集合。对于事实表,其粒度包括所有相关联的维度键。例如,在图4中,Store_Returns的粒度为Date_key、Item_key、Store_key、Customer_key。另外,对于每个维度,由于所有属性在功能上都可以由键属性确定,因此它们在事实表的粒度中有效。因此,例如,Year、Quarter、Month等可由Date_key确定,因此它们在Store_Returns粒度中有效。应用类似的自变量,图4中的所有维度列在Store_Returns表的粒度中均有效。

[0211] 现在在图3中,聚合器操作/运算符324按照Customer_key、Store_key输出行,因此由操作324定义的数据集的粒度为Store_key、Customer_key。但是再次,通过功能依赖性,所有Customer和Store列在其粒度中均有效。因此,对于事实表Store_Returns,在图4中所描绘的星形架构上定义的具有架构(Cust_Name,Item_Category,Store_Name,Qty)的数据集可以具有粒度(Cust_Name,Item_Category,Store_Name)。

[0212] 在遍历开始时,用于事实扫描操作的事实表的粒度是事实表的相关联维度中的所有属性。因此,在图3中,对于子计划302中的Store_Returns,这将是来自Date、Item、Store和Customer维度表中的所有列,还包括Customer_Address维度表中的辅助Customer属性。这基于为图4中的事实表定义的星形架构。

[0213] 在1324中的遍历期间,在遇到与联接操作对应的节点后执行特殊处理。这涉及捕获与事实表的确认联接并尝试推断可以从联接的“另一侧”传播到事实扫描操作侧或“事实侧”的适用上下文。例如,在图3中,对于联接操作节点322,事实侧将是Store_Returns节点320,而另一侧将是过滤操作节点328。作为另一个示例,在图3中,对于节点306,事实侧将是左子计划302,而另一侧将是右子计划312。

[0214] 确认联接意味着该联接不改变事实侧的粒度。例如,最常见的确认模式之一涉及与星形架构中的边上的条件相同的联接条件。因而,对于涉及图4中所描绘的星形架构的查询,联接必须在Store_Returns和Store之间在store_key上,或者Store_Returns和customer_key等等。相反,非确认联接的示例将包括Store_Returns物化(诸如物化的视图或OLAP索引)与Customer之间在customer_name上的联接或Store_Returns与Store_Rent表之间在store_name上的联接;在此Store_Rent不是星形架构的一部分。

[0215] 在遇到确认联接后,该推断逻辑然后在确认联接时检查以下附加条件:事实侧粒度必须包括正被联接的维度粒度。如果是这种情况,那么联接的另一侧上由联接粒度在功能上确定的任何上下文(谓词条件)都适用于事实侧,因此适用于事实表。例如,对于图3中所描绘的查询计划,对于最左侧的“在Date Key上联接”节点322,联接运算符的左事实侧子树具有粒度“Customer、Item、Date、Store”,另一侧上的联接在“Date”维度上;因此,这是在Date Key粒度的确认联接。所有Date属性都可以由Date Key确定,因此关于Year的谓词可以传播到由节点320表示的事实表Store_Returns上的事实扫描操作。可以对子计划312中的Store_Returns上的事实扫描操作执行类似的分析。

[0216] 关于适用的粒度的信息在遍历期间被维护和更新。首先,识别粒度中的所有维度(Dimension)属性。在遍历聚合操作/运算符(例如,图3中的节点324和330)时,粒度改变为

分组列。例如,在图3中,在遍历节点324中的聚合操作/运算符时,粒度变为(Customer_key,Store_key)。在某些实施例中,关于遍历在查询计划中发现的不同类型的关系操作/运算符,配置和维护如何修改和维护粒度的规则。

[0217] 作为1324中的遍历的一部分,对于每个事实表扫描运算符,信息由增强器103维护,包括与所涉及的联接表的确认联接的列表、已经为那个事实扫描操作识别或推断出的适用的一个或多个维度上下文谓词条件的列表。在事实扫描操作的遍历结束时,正被遍历的事实扫描运算符具有适用的一个或多个维度上下文(谓词条件)和确认联接的候选列表。

[0218] 在1326处,执行谓词条件的传递式事实到事实推断。这个处理涉及对跨相同事实或不同事实的谓词条件的传递式推断。对于每对事实扫描运算符,比如说“fs1”(事实表ft1上的事实扫描)和“fs2”(事实表ft2上的事实扫描),增强器103寻找联接运算符,以使联接条件在维度D1上具有联接属性D1.ja,使得D1处于ft1和ft2两者的星形架构中。

[0219] 例如,在图12B中:

[0220] -fs1是Store_Returns扫描(方框1224);

[0221] -fs2是Store_Sales扫描(方框1226);

[0222] -联接两个事实表的联接是节点1232;

[0223] -公共维度D1是Store维度;以及

[0224] -联接是在Store的Store_key属性上。

[0225] 在找到这种模式后,如果所涉及的两个事实表都在联接属性D1.ja的粒度上,那么增强器103可以在维度D1上传播任何推断的维度上下文,维度D1由fs1到fs2的联接粒度在功能上确定,反之亦然。因而,在图12B中,来自Store_Returns扫描(方框1224)的推断出的上下文谓词条件“State＝TN”可以经由方框1232中的Store_key上的联接传播到Store_Sales扫描(节点1226)。这种事实到事实的推断是传递的;因此维度上下文可以从fs1(在ft1上)传播到fs2(在ft2上)传播到f3(在ft3上),依此类推。

[0226] 在1328处,执行谓词条件的传递式事实到维度推断。由于星形架构可以是雪花模式,因此星形架构中从维度表到事实表的联接路径的长度可以大于一。可以使用图17中描绘的架构1700来描述1328中的处理。在图17中,Store_Returns是事实表,其它表都是维度表(注意的是,架构1700是图4中描绘的架构400的细微变化)。在图4中,Income_Band仅通过四表联接与Store_Returns相关联:Income_Band联接Customer_Demographics联接Customer联接Store_Returns。如果存在包含Income_Band属性的Store_Returns的物理表示,例如Store_Returns上包含income_upper_bound属性的OLAP索引,那么可以在其上应用Income_upper_bound谓词。为了让增强器103推断Income_upper_bound上下文(谓词)在OLAP索引事实源上的应用,它必须通过将Income_Band维度表与事实表连接的三联接(Income_Band联接Customer_Demographic,Customer_Demographic联接Customer,以及Customer联接Store_returns)来传播。通过递归建立事实扫描运算符的确认联接的列表来完成这个操作,并且在每次遇到新的适用的确认联接时,都会尝试从另一侧传播任何可用的维度上下文,如1324中所示。

[0227] 在1330处,一个或多个推断的维度上下文谓词条件被应用并且与事实扫描操作相关联。到此为止,在适用的情况下,原始查询计划中识别出的每个事实表扫描操作/运算符(或一般而言,事实源扫描操作)都与一个或多个适用的维度上下文谓词条件的列表或集合

相关联。作为1330中处理的一部分,为每个事实扫描操作识别出的谓词条件将根据其净收益进行排序,其中谓词条件的净收益是在应用谓词条件后处理事实行的成本减去应用谓词条件的成本。

[0228] 因而,作为1330的一部分,对于每个事实扫描操作,增强器103可以为针对那个事实扫描操作确定的适用谓词条件列表中的每个谓词条件计算净收益度量。增强器103可以考虑若干因素来计算净收益度量并执行排序,诸如:

[0229] -事实的可用物理表示,诸如对事实的OLAP索引以及OLAP索引中的列的可用性。注意的是,并非所有维度列都需要在OLAP索引中。

[0230] -推断出的维度上下文的选择性;高选择性谓词优于低选择性谓词。

[0231] -应用维度上下文谓词条件的成本。例如,将谓词推送到OLAP索引扫描几乎不会招致任何附加成本,因此是优选的;

[0232] 索引半联接优于传统的半联接,因为索引半联接充分利用在索引扫描操作/运算符中执行高效跳过扫描的能力。

[0233] 作为1330的一部分,在排序之后,对于每个事实扫描操作,从谓词条件的列表中确定一个或多个谓词条件,以基于该排序与事实扫描相关联。在某些实施例中,对于事实扫描操作,来自列表的提供最佳净正收益的谓词条件被选择为与该事实扫描操作相关联。

[0234] 在某些实施例中,对于事实扫描操作,来自该事实扫描操作的谓词条件的有序列表中的多于一个谓词条件可以与该事实扫描操作相关联。在这种场景中,在提供最高净收益的谓词条件已经与事实扫描操作相关联之后,对于有序列表中的每个附加谓词条件,从提供次佳净收益的谓词条件开始,考虑到那个事实扫描操作的那个附加谓词条件和所有先前相关联的谓词条件的关联而进行净收益分析,以确定附加谓词条件是否要与事实扫描操作相关联。当多个谓词条件与事实扫描操作相关联时,应用附加谓词条件的收益通常随着相关联的谓词条件数量而减少。

[0235] 在1330中的处理之后,对于特定的事实扫描操作,有可能出现被识别为适用于那个事实扫描操作的一个或多个谓词条件中的任何一个都不提供净收益。在这种情况下,没有谓词条件可以与那个事实扫描操作相关联。

[0236] 在1332中,增强器103执行处理,以确定是否可以通过基于与事实扫描操作相关联的现有谓词条件推断查询计划中一个或多个事实扫描操作的任何新的一个或多个谓词条件来进一步优化查询计划的执行。事实扫描操作的现有谓词条件可以是与原始查询计划中的事实扫描操作相关联的谓词条件(原始谓词条件)和/或在1330中执行的处理之后与事实扫描操作相关联的谓词条件。因而,作为在1332中执行的处理的一部分,如果新谓词条件比原始谓词条件为事实扫描操作提供更大的净收益,那么增强器103可以尝试基于与事实扫描操作相关联的原始谓词条件来找到新的经转化的谓词条件。以类似的方式,对于与1330中的事实扫描操作相关联的谓词条件,如果新谓词条件比与1330中的事实扫描操作相关联的谓词条件为事实扫描操作提供更大的净收益,那么增强器103可以尝试基于相关联的谓词条件来找到新的经转化的谓词条件。

[0237] 在某些实施例中,作为1322的一部分,增强器103尝试基于维度属性之间的功能依赖性从识别出/推断出的谓词中找到隐式谓词。例如,对于图3中所描绘的查询计划,我们在图5的方框502中示出对左侧Store_Returns应用谓词条件“State＝TN”。假定“State＝TN”

的这个应用不是成本有效了,可能是因为可用的OLAP索引不具有State属性。在这种情况下,增强器103可以使用以下元数据信息来推断新谓词条件:

[0238] -State和City列在层次结构中;

[0239] -“TN”仅具有在数据库中列出的2个城市;以及

[0240] -City列在OLAP索引中。

[0241] 以上信息可以被用来有可能推断“State＝TN”维度上下文可以应用为“City＝Memphis or City＝Nashville”,其中推断出的谓词条件的应用涉及非常快的OLAP索引扫描。应用这个推断出的谓词条件可以向执行总体增强的查询计划提供净收益。经转化的谓词条件的另一个示例在图9中描绘并在上面进行了描述。

[0242] 在某些实施例中,仅在以下情况下对查询计划中的事实扫描操作执行1322中的处理:不存在与事实扫描操作相关联的原始谓词条件(即在原始查询计划中);基于在1324、1326和1328中执行的处理,在有资格与事实扫描操作相关联的谓词条件的列表中识别出了至少一个谓词条件;以及,基于在1330中执行的处理,确定为事实扫描操作确定的列表中的谓词条件均未提供任何正净收益,因此不与1330中的事实扫描操作相关联。在这种场景中,增强器103可以在针对事实扫描操作识别出的谓词条件的列表中的一个或多个谓词条件上执行1332中的处理,以查看是否有任何这样的经转化的新谓词条件为事实扫描操作提供正净收益。如果找到了这样的经转化的新谓词条件,那么它可以在1332由增强器103与事实扫描操作相关联。但是,这并不旨在是限制性的。在一些其它实施例中,可以针对与1330中的事实扫描操作相关联的任何原始谓词条件和/或任何谓词条件执行1332中的处理。

[0243] 在1334处,基于原始查询计划生成增强的查询计划,该查询计划具有与一个或多个事实扫描操作相关联的一个或多个谓词条件。在某些实施例中,原始查询计划被重写为增强的查询计划,使得在增强的查询计划中被识别为与一个或多个事实扫描操作相关联的一个或多个谓词条件实际上与事实扫描操作相关联。因而,当作为执行查询的一部分在1308中执行增强的查询计划时,针对它们与之相关联的事实扫描操作评估相关联的谓词条件。

[0244] 基于本文公开的教导而生成的增强的查询计划提供了优于常规技术的若干益处。如上所述,诸如SQL查询之类的分析查询通常涉及多个维度的计算,并且需要对通过导航维度关联进行组合的一个或多个事实表进行多层计算。作为示例,维度关联的常见示例是层次结构,如Year-Quarter-Month或Country-State-City等。在图3中所描绘的示例中,子计划302涉及计算Customers在每个Store的Store Returns,并且子计划312涉及在Store处对Store的所有Customers的聚合退货的计算。在节点306处,将这两者组合以发现“问题”客户。这种兆兆字节(或甚至更高)规模和数据库之外的多层计算的计算量非常大,并且要在“思考时间”内回答任意查询非常具有挑战性。本文公开的各种发明技术提供了一种分析平台,该分析平台在包括但不限于OLAP加索引的这种规模上使得能够对自组织查询有交互式体验,并且横向扩展无共享架构。本文描述的分析平台提供了以非常成本有效的方式在非常大(兆兆字节及以上)的多维数据(例如,存储在数据湖中的数据)上运行交互式自组织查询的能力;过去使用常规技术在非常大(兆兆字节及以上)的多维数据上运行这种交互式自组织查询非常昂贵而且无法扩展。

[0245] 对于查询而言,分析师的目的常常是集中于广阔的多维空间的一小个区域。例如

在图3中,分析的重点是Tennessee州。但是对于像SQL这样的查询语言,由于使用关系代数运算符编写此类多维计算会是笨拙的,因此这种分析上下文会变得晦涩难懂。进一步的SQL编写常常是通过SQL Generator工具或不是作为领域专家的数据工程师完成的,它们的目标是产生给出正确答案的SQL;他们对以如下方式、就像对“Tennessee州”的分析那样携带语义信息不感兴趣,该方式使得SQL查询引擎易于生成避免不必要的事实处理的查询计划。常规的关系查询优化器在关系模式上运行,因此缺乏关于非常有价值的多维立方体语义模型的知识,多维立方体语义模型是如何表达分析问题的基础;每个分析师的脑海中都会浮现出这种模型。本文公开的分析平台执行处理,以通过充分利用多维立方体语义模型中定义的关系,从查询计划的结构中对分析的上下文进行逆向工程。在许多情况下,这导致推断显著减少处理事实所需的资源的适用的维度上下文(谓词条件)。在大数据规模(例如,兆兆字节规模)及以上,如本文所述,事实处理成本的降低对交互式处理自组织查询的能力具有重大影响。

[0246] 如上所述,提供了一种分析平台,用于使分析查询更高效地执行。在某些实施例中,提供了Apache Spark原生分析平台。针对分析查询(例如,SQL查询)生成的查询计划被优化以创建优化和增强的查询计划。语义信息(诸如商业数据和商业智能数据)被用于生成增强的查询计划。领域语义被用于生成更高效的查询计划。商业语义被用于查询执行优化。

[0247] 从性能的角度来看,增强的查询计划比未增强的查询计划执行得更快(因为它们需要较少的CPU周期来完成其执行),并且在许多情况下可以使用更少的存储器资源。作为重写的查询计划的结果,对应的查询可以比重写之前执行得快一个数量级(例如,10X、25X、100X等)。这是因为事实表比维度表大几个数量级,因此减少与处理事实记录相关的扫描成本带来巨大的性能提升。

[0248] 如前面所指示的,本文描述的分析平台可以使用Spark集群实现为Apache Spark原生方案。因此,它充分利用了Apache Spark及其生态系统的功能,如核心组件、可伸缩的存储器运行时、催化剂组件、JDBC、元存储库等。

[0249] 如本文所述,描述了用于生成优化的查询计划的技术,其中与未增强的查询计划相比,减少了通过增强的查询计划处理和扫描事实记录的成本。分析活动常常专注于特定的子群体或子上下文,而不是整个数据集。从语义信息推断适用于查询的上下文并将上下文应用于查询计划(例如,通过上下文或谓词条件传播)可以导致查询计划以及因此为其生成查询计划的查询的执行的显著性能提高。无论是分析领域还是分析活动,都有可以被识别出以确定分析的上下文的通用模式。在某些实施例中,提供了用于识别这些模式并生成优化的增强查询计划的技术。

[0250] 图14描绘了用于实现实施例的分布式系统1400的简化图。在所示的实施例中,分布式系统1400包括经由一个或多个通信网络1410耦合到服务器1412的一个或多个客户端计算设备1402、1404、1406和1408。客户端计算设备1402、1404、1406和1408可以被配置为执行一个或多个应用。

[0251] 在各种实施例中,服务器1412可以适于运行使得能够执行如上所述的分析查询的一个或多个服务或软件应用。

[0252] 在某些实施例中,服务器1412还可以提供可以包括非虚拟和虚拟环境的其它服务或软件应用。在一些实施例中,这些服务可以作为基于web的服务或云服务,诸如在软件即

服务(SaaS)模型下,提供给客户端计算设备1402、1404、1406和/或1408的用户。操作客户端计算设备1402、1404、1406和/或1408的用户可以依次利用一个或多个客户端应用与服务器1412交互以利用由这些组件提供的服务。

[0253] 在图14所描绘的配置中,服务器1412可以包括实现由服务器1412执行的功能的一个或多个组件1418、1420和1422。这些组件可以包括可以由一个或多个处理器执行的软件组件、硬件组件或其组合。应当认识到的是,各种不同的系统配置是可能的,其可以与分布式系统1400不同。因此,图14中所示的实施例是用于实现实施例系统的分布式系统的一个示例,并且不旨在进行限制。

[0254] 根据本公开的教导,用户可以使用客户端计算设备1402、1404、1406和/或1408提交分析查询。客户端设备可以提供使客户端设备的用户能够与客户端设备交互的界面。客户端设备还可以经由这个界面向用户输出信息。例如,查询结果可以经由客户端设备提供的界面被输出给用户。虽然图14仅描绘了四个客户端计算设备,但可以支持任何数量的客户端计算设备。

[0255] 客户端设备可以包括各种类型的计算系统,诸如便携式手持式设备、通用计算机(诸如个人计算机和膝上型计算机)、工作站计算机、可穿戴设备、游戏系统、瘦客户端、各种消息传送设备、传感器和其它感测设备等。这些计算设备可以运行各种类型和版本的软件应用和操作系统(例如,Microsoft **Windows®**、Apple **Macintosh®**、**UNIX®**或类UNIX操作系统、Linux或类Linux操作系统(诸如Google Chrome™ OS)),包括各种移动操作系统(例如,Microsoft Windows **Mobile®** 、 **iOS®** 、Windows **Phone®** 、Android™、**BlackBerry®**、Palm **OS®**)。便携式手持式设备可以包括蜂窝电话、智能电话(例如,**iPhone®**)、平板电脑(例如,**iPad®**)、个人数字助理(PDA)等。可穿戴设备可以包括Google **Glass®**头戴式显示器和其它设备。游戏系统可以包括各种手持式游戏设备、具有互联网功能的游戏设备(例如,具有或不具有**Kinect®**手势输入设备的Microsoft **Xbox®**游戏机、Sony **PlayStation®**系统、由**Nintendo®**提供的各种游戏系统以及其它)等。客户端设备可以能够执行各种不同的应用,诸如各种与互联网相关的应用、通信应用(例如,电子邮件应用、短消息服务(SMS)应用),并且可以使用各种通信协议。

[0256] (一个或多个)通信网络1410可以是本领域技术人员熟悉的任何类型的网络,其可以使用多种可用协议中的任何一种来支持数据通信,包括但不限于TCP/IP(传输控制协议/互联网协议)、SNA(系统网络架构)、IPX(互联网分组交换)、**AppleTalk®**等。仅仅作为示例,(一个或多个)网络1410可以是局域网(LAN)、基于以太网的网络、令牌环、广域网(WAN)、互联网、虚拟网络、虚拟专用网(VPN)、内联网、外联网、公共电话交换网(PSTN)、红外网络、无线网络(例如,在任何电气和电子协会(IEEE)1002.11协议套件、**蓝牙®**和/或任何其它无线协议下操作的网络)和/或这些和/或其它网络的任意组合。

[0257] 服务器1412可以包括一个或多个通用计算机、专用服务器计算机(作为示例,包括PC(个人计算机)服务器、**UNIX®**服务器、中档服务器、大型计算机、机架安装的服务器等)、服务器场、服务器集群或任何其它适当的布置和/或组合。服务器1412可以包括运行虚

拟操作系统的一个或多个虚拟机,或者涉及虚拟化的其它计算架构,诸如可以被虚拟化以维护服务器的虚拟存储设备的逻辑存储设备的一个或多个灵活的池。在各种实施例中,服务器1412可以适于运行提供前述公开中描述的功能的一个或多个服务或软件应用。

[0258] 服务器1412中的计算系统可以运行一个或多个操作系统,包括以上讨论的任何操作系统以及任何商用的服务器操作系统。服务器1412还可以运行各种附加服务器应用和/或中间层应用中的任何一种,包括HTTP(超文本传输协议)服务器、FTP(文件传输协议)服务器、CGI(通用网关接口)服务器、**JAVA®**服务器、数据库服务器等。示例性数据库服务器包括但不限于可从**Oracle®、 Microsoft®、Sybase®、 IBM®** (国际商业机器)等商购获得的数据库服务器。

[0259] 在一些实施方式中,服务器1412可以包括一个或多个应用以分析和整合从客户端计算设备1402、1404、1406和1408的用户接收到的数据馈送和/或事件更新。作为示例,数据馈送和/或事件更新可以包括但不限于从一个或多个第三方信息源和连续数据流接收到的**Twitter®**馈送、**Facebook®**更新或实时更新,其可以包括与传感器数据应用、金融报价机、网络性能测量工具(例如,网络监视和流量管理应用)、点击流分析工具、汽车流量监视等相关的实时事件。服务器1412还可以包括经由客户端计算设备1402、1404、1406和1408的一个或多个显示设备显示数据馈送和/或实时事件的一个或多个应用。

[0260] 分布式系统1400还可以包括一个或多个数据储存库1414、1416。在某些实施例中,这些数据储存库可以用于存储数据和其它信息。例如,一个或多个数据储存库1414、1416可以用于存储数据、元数据等。数据储存库1414、1416可以驻留在各种位置。例如,由服务器1412使用的数据储存库可以在服务器1412本地,或者可以远离服务器1412并经由基于网络或专用的连接与服务器1412通信。数据储存库1414、1416可以是不同的类型。在某些实施例中,由服务器1412使用的数据储存库可以是数据库,例如关系数据库,诸如由**Oracle®**公司和其它供应商提供的数据库。这些数据库中的一个或多个可以适于响应于SQL格式的命令来实现数据到数据库或从数据库的存储、更新和检索。

[0261] 在某些实施例中,应用还可以使用数据储存库1414、1416中的一个或多个来存储应用数据。由应用使用的数据储存库可以具有不同的类型,诸如,例如,键值储存库、对象储存库或由文件系统支持的通用储存库。

[0262] 在某些实施例中,可以经由云环境将本公开中描述的分析平台提供的功能提供为服务。图15是根据某些实施例的其中各种服务可以被提供为云服务的基于云的系统环境的简化框图。在图15所示的实施例中,云基础设施系统1502可以提供可以由用户使用一个或多个客户端设备1504、1506和1508请求的一个或多个云服务。云基础设施系统1502可以包括一个或多个计算机和/或服务器,其可以包括以上针对服务器1412描述的那些服务器。云基础设施系统1502中的计算机可以被组织为通用计算机、专用服务器计算机、服务器场、服务器集群或任何其它适当的布置和/或组合。

[0263] (一个或多个)网络1510可以促进客户端设备1504、1506和1508与云基础设施系统1502之间的通信和数据交换。(一个或多个)网络1510可以包括一个或多个网络。网络可以是相同或不同的类型。(一个或多个)网络1510可以支持一种或多种通信协议(包括有线和/或无线协议),以促进通信。

[0264] 图15中描绘的实施例仅仅是云基础设施系统的一个示例,并且不旨在进行限制。应该认识到的是,在一些其它实施例中,云基础设施系统1502可以具有比图15中所示的组件更多或更少的组件,可以组合两个或更多个组件,或者可以具有不同的组件配置或布置。例如,虽然图15描绘了三个客户端计算设备,但是在替代实施例中可以支持任何数量的客户端计算设备。

[0265] 术语“云服务”通常用于指由服务提供商的系统(例如,云基础设施系统1502)根据需要并且经由诸如互联网之类的通信网络使得用户可使用的服务。典型地,在公共云环境中,组成云服务提供商的系统的服务器和系统与客户自己的内部部署服务器和系统不同。云服务提供商的系统由云服务提供商管理。客户因此可以利用由云服务提供商提供的云服务,而不必为服务购买单独的许可证、支持或硬件和软件资源。例如,云服务提供商的系统可以托管应用,并且用户可以经由互联网按需订购和使用应用,而用户不必购买用于执行应用的基础设施资源。云服务旨在提供对应用、资源和服务的轻松、可扩展的访问。几个提供商提供云服务。例如,由加利福尼亚州Redwood Shores的Oracle®公司提供了几种云服务,诸如中间件服务、数据库服务、Java云服务等。

[0266] 在某些实施例中,云基础设施系统1502可以使用诸如软件即服务(SaaS)模型、平台即服务(PaaS)模型、基础设施即服务(IaaS)模型以及包括混合服务模型的其它模型之类的不同模型提供一个或多个云服务。云基础设施系统1502可以包括一套应用、中间件、数据库以及使得能够供给各种云服务的其它资源。

[0267] SaaS模型使得应用或软件能够作为服务通过如互联网的通信网络交付给客户,而客户不必为底层应用购买硬件或软件。例如,SaaS模型可以用于为客户提供对由云基础设施系统1502托管的按需应用的访问。由Oracle®公司提供的SaaS服务的示例包括但不限于用于人力资源/资本管理、客户关系管理(CRM)、企业资源计划(ERP)、供应链管理(SCM)、企业绩效管理(EPM)、分析服务、社交应用及其它的各种服务。

[0268] IaaS模型通常用于向客户提供基础设施资源(例如,服务器、存储装置、硬件和联网资源)作为云服务,以提供弹性计算和存储能力。Oracle®公司提供了各种IaaS服务。

[0269] PaaS模型通常用于提供平台和环境资源作为服务,其使得客户能够开发、运行和管理应用和服务,而客户不必采购、构建或维护此类资源。由Oracle®公司提供的PaaS服务的示例包括但不限于Oracle Java云服务(JCS)、Oracle数据库云服务(DBCS)、数据管理云服务、各种应用开发方案服务以及其它服务。

[0270] 云服务通常基于按需自服务、基于订阅、弹性可缩放、可靠、高度可用和安全的方式提供。例如,客户可以经由订阅订单订购由云基础设施系统1502提供的一个或多个服务。然后,云基础设施系统1502执行处理,以提供客户的订阅订单中所请求的服务。例如,系统1502可以服务于本公开中所描述的分析查询。云基础设施系统1502可以被配置为提供一个或甚至多个云服务。

[0271] 云基础设施系统1502可以经由不同的部署模型来提供云服务。在公共云模型中,云基础设施系统1502可以由第三方云服务提供商拥有,并且云服务被提供给任何普通公众客户,其中客户可以是个人或企业。在某些其它实施例中,在私有云模型下,可以在组织内(例如,在企业组织内)操作云基础设施系统1502,并向组织内的客户提供服务。例如,客户

可以是企业的各个部门,诸如人力资源部门、工资部门等,甚至是企业内的个人。在某些其它实施例中,在社区云模型下,云基础设施系统1502和所提供的服务可以由相关社区中的几个组织共享。也可以使用各种其它模型,诸如上面提到的模型的混合。

[0272] 客户端设备1504、1506和1508可以是不同类型的(诸如图14中描绘的客户端设备1404、1406和1408),并且可以能够操作一个或多个客户端应用。用户可以使用客户端设备与云基础设施系统1502交互,诸如请求由云基础设施系统1502提供的服务。

[0273] 在一些实施例中,由云基础设施系统1502执行的用于提供云服务的处理可能涉及大数据分析。这种分析可能涉及使用、分析和操纵大型数据集,以检测和可视化数据内的各种趋势、行为、关系等。该分析可以由一个或多个处理器执行、可能并行处理数据、使用数据执行仿真等。用于该分析的数据可以包括结构化数据(例如,存储在数据库中或根据结构化模型结构化的数据)和/或非结构化数据(例如,数据blob(二进制大对象))。

[0274] 如在图15的实施例中所描绘的,云基础设施系统1502可以包括基础设施资源1530,其用于促进由云基础设施系统1502提供的各种云服务的供给。基础设施资源1530可以包括例如处理资源、存储或存储器资源、联网资源等。

[0275] 在某些实施例中,为了促进这些资源的高效供给以支持由云基础设施系统1502为不同客户提供的各种云服务,可以将资源捆绑成资源或资源模块的集合(也称为“群聚(pod)”)。每个资源模块或群聚可以包括一种或多种类型的资源的预先集成和优化的组合。在某些实施例中,可以为不同类型的云服务预先供给不同的群聚。例如,可以为数据库服务供给第一群聚集合,可以为Java服务供给可以包括与第一群聚集合中的群聚不同的资源组合的第二群聚集合,等等。对于一些服务,可以在服务之间共享为供给服务而分配的资源。

[0276] 云基础设施系统1502本身可以内部使用服务1532,服务1532由云基础设施系统1502的不同组件共享并且促进云基础设施系统1502的服务供给。这些内部共享的服务可以包括但不限于安全和身份服务、集成服务、企业储存库服务、企业管理器服务、病毒扫描和白名单服务、高可用性、备份和恢复服务、用于启用云支持的服务、电子邮件服务、通知服务、文件传输服务等。

[0277] 云基础设施系统1502可以包括多个子系统。这些子系统可以用软件或硬件或其组合来实现。如图15中所示,子系统可以包括用户界面子系统1512,该用户界面子系统1512使得云基础设施系统1502的用户或客户能够与云基础设施系统1502交互。用户界面子系统1512可以包括各种不同的界面,诸如web界面1514、在线商店界面1516(其中由云基础设施系统1502提供的云服务被广告并且可由消费者购买)和其它界面1518。例如,客户可以使用客户端设备使用界面1514、1516和1518中的一个或多个来请求(服务请求1534)由云基础设施系统1502提供的一个或多个服务。例如,客户可以访问在线商店、浏览由云基础设施系统1502提供的云服务,并为客户希望订阅的由云基础设施系统1502提供的一项或多项服务下订阅订单。服务请求可以包括识别客户的信息以及客户期望订阅的一项或多项服务。

[0278] 在某些实施例中,诸如图15中所示的实施例,云基础设施系统1502可以包括被配置为处理新的订单的订单管理子系统(OMS)1520。作为该处理的一部分,OMS1520可以被配置为:为客户创建账户(如果尚未创建);接收来自客户的账单和/或会计信息,该账单和/或账单信息将用于针对向客户提供所请求的服务向客户计费;核实客户信息;在核实后,为客户预订订单;编排各种工作流程以准备用于供给的订单。

[0279] 一旦被正确地证实,OMS1520就可以调用订单供给子系统(OPS)1524,其被配置为为订单供给资源,包括处理资源、存储器资源和联网资源。供给可以包括为订单分配资源,以及配置资源以促进由客户订单所请求的服务。为订单供给资源的方式和供给资源的类型可以取决于客户已订购的云服务的类型。例如,根据一个工作流程,OPS1524可以被配置为确定所请求的特定云服务,并且识别可能已经针对该特定云服务而被预先配置的多个群聚。为订单分配的群聚的数量可以取决于所请求的服务的大小/数量/级别/范围。例如,可以基于服务所支持的用户的数量、正在请求的服务的持续时间等来确定要分配的群聚的数量。然后,可以针对特定的请求客户定制所分配的群聚,以提供所请求的服务。

[0280] 云基础设施系统1502可以向请求客户发送响应或通知1544,以指示所请求的服务何时准备就绪。在一些情况下,可以将信息(例如,链接)发送给客户,使得客户能够开始使用和利用所请求的服务的益处。

[0281] 云基础设施系统1502可以向多个客户提供服务。对于每个客户,云基础设施系统1502负责管理与从客户接收到的一个或多个订阅订单相关的信息,维护与订单相关的客户数据,以及向客户提供所请求的服务。云基础设施系统1502还可以收集关于客户对已订阅的服务的使用的使用统计信息。例如,可以针对使用的存储量、传输的数据量、用户的数量以及系统正常运行时间量和系统停机时间量等收集统计信息。该使用信息可以用于向客户计费。计费可以例如按月周期进行。

[0282] 云基础设施系统1502可以并行地向多个客户提供服务。云基础设施系统1502可以存储这些客户的信息,包括可能的专有信息。在某些实施例中,云基础设施系统1502包括身份管理子系统(IMS)1528,其被配置为管理客户的信息并提供所管理的信息的分离,使得与一个客户相关的信息不可被另一个客户访问。IMS1528可以被配置为提供各种与安全相关的服务,诸如身份服务,诸如信息访问管理、认证和授权服务、用于管理客户身份和角色及相关能力的服务等等。

[0283] 图16图示了可以用于实现某些实施例的示例性计算机系统1600。例如,在一些实施例中,计算机系统1600可以用于实现图1A和图1B中描绘的任何系统。如图16中所示,计算机系统1600包括各种子系统,包括经由总线子系统1602与多个其它子系统通信的处理子系统1604。这些其它子系统可以包括处理加速单元1606、I/O子系统1608、存储子系统1618和通信子系统1624。存储子系统1618可以包括非暂态计算机可读存储介质,其包括存储介质1622和系统存储器1610。

[0284] 总线子系统1602提供用于使计算机系统1600的各种组件和子系统按照期望彼此通信的机制。虽然总线子系统1602被示意性地示为单条总线,但是总线子系统的替代实施例可以利用多条总线。总线子系统1602可以是几种类型的总线结构中的任何一种,包括存储器总线或存储器控制器、外围总线、使用任何各种总线架构的局部总线等。例如,此类架构可以包括工业标准架构(ISA)总线、微通道架构(MCA)总线、增强型ISA(EISA)总线、视频电子标准协会(VESA)局部总线和外围组件互连(PCI)总线,其可以实现为根据IEEE P1386.1标准制造的夹层(Mezzanine)总线等等。

[0285] 处理子系统1604控制计算机系统1600的操作,并且可以包括一个或多个处理器、专用集成电路(ASIC)或现场可编程门阵列(FPGA)。处理器可以包括单核或多核处理器。可以将计算机系统1600的处理资源组织成一个或多个处理单元1632、1634等。处理单元可以

包括一个或多个处理器、来自相同或不同处理器的一个或多个核、核和处理器的组合、或核和处理器的其它组合。在一些实施例中,处理子系统1604可以包括一个或多个专用协处理器,诸如图形处理器、数字信号处理器(DSP)等。在一些实施例中,处理子系统1604的一些或全部可以使用定制电路来实现,诸如专用集成电路(ASIC)或现场可编程门阵列(FPGA)。

[0286] 在一些实施例中,处理子系统1604中的处理单元可以执行存储在系统存储器1610中或计算机可读存储介质1622上的指令。在各个实施例中,处理单元可以执行各种程序或代码指令,并且可以维护多个并发执行的程序或进程。在任何给定的时间,要执行的程序代码中的一些或全部可以驻留在系统存储器1610中和/或计算机可读存储介质1622上,包括可能在一个或多个存储设备上。通过适当的编程,处理子系统1604可以提供上述各种功能。在计算机系统1600正在执行一个或多个虚拟机的情况下,可以将一个或多个处理单元分配给每个虚拟机。

[0287] 在某些实施例中,可以可选地提供处理加速单元1606,以用于执行定制的处理或用于卸载由处理子系统1604执行的一些处理,从而加速由计算机系统1600执行的整体处理。

[0288] I/O子系统1608可以包括用于向计算机系统1600输入信息和/或用于从或经由计算机系统1600输出信息的设备和机制。一般而言,术语“输入设备”的使用旨在包括用于向计算机系统1600输入信息的所有可能类型的设备和机制。用户界面输入设备可以包括,例如,键盘、诸如鼠标或轨迹球之类的指向设备、并入到显示器中的触摸板或触摸屏、滚轮、点击轮、拨盘、按钮、开关、小键盘、带有语音命令识别系统的音频输入设备、麦克风以及其它类型的输入设备。用户界面输入设备还可以包括使用户能够控制输入设备并与之交互的诸如Microsoft **Kinect®**运动传感器的运动感测和/或姿势识别设备、Microsoft **Xbox®** 360游戏控制器、提供用于接收使用姿势和口语命令的输入的界面的设备。用户界面输入设备还可以包括眼睛姿势识别设备,诸如从用户检测眼睛活动(例如,当拍摄图片和/或进行菜单选择时的“眨眼”)并将眼睛姿势转换为到输入设备(例如,Google **Glass®**)的输入的Google **Glass®**眨眼检测器。此外,用户界面输入设备可以包括使用户能够通过语音命令与语音识别系统(例如,**Siri®**导航器)交互的语音识别感测设备。

[0289] 用户界面输入设备的其它示例包括但不限于,三维(3D)鼠标、操纵杆或指示杆、游戏板和图形平板、以及音频/视频设备,诸如扬声器、数字相机、数字摄像机、便携式媒体播放器、网络摄像机、图像扫描仪、指纹扫描仪、条形码读取器3D扫描仪、3D打印机、激光测距仪、以及眼睛注视跟踪设备。此外,用户界面输入设备可以包括,例如,医疗成像输入设备,诸如计算机断层摄影、磁共振成像、位置发射断层摄影、以及医疗超声检查设备。用户界面输入设备也可以包括,例如音频输入设备,诸如MIDI键盘、数字乐器等。

[0290] 一般而言,术语“输出设备”的使用旨在包括所有可能类型的设备和用于从计算机系统1600向用户或其它计算机输出信息的机制。用户界面输出设备可以包括显示子系统、指示器灯或诸如音频输出设备的非可视显示器等。显示子系统可以是阴极射线管(CRT)、诸如利用液晶显示器(LCD)或等离子体显示器的平板设备、投影设备、触摸屏等。例如,用户界面输出设备可以包括但不限于,可视地传达文本、图形和音频/视频信息的各种显示设备,诸如监视器、打印机、扬声器、耳机、汽车导航系统、绘图仪、语音输出设备和调制解调器。

[0291] 存储子系统1618提供用于存储由计算机系统1600使用的信息和数据的储存库或数据存储库。存储子系统1618提供用于存储提供某些实施例的功能的基本编程和数据构造的有形非暂态计算机可读存储介质。存储子系统1618可以存储软件(例如,程序、代码模块、指令),该软件在由处理子系统1604执行时提供上述功能。软件可以由处理子系统1604的一个或多个处理单元执行。存储子系统1618还可以提供用于存储根据本公开的教导使用的数据的储存库。

[0292] 存储子系统1618可以包括一个或多个非暂态存储器设备,包括易失性和非易失性存储器设备。如图16中所示,存储子系统1618包括系统存储器1610和计算机可读存储介质1622。系统存储器1610可以包括多个存储器,包括用于在程序执行期间存储指令和数据的易失性主随机存取存储器(RAM)以及其中存储有固定指令的非易失性只读存储器(ROM)或闪存。在一些实现中,基本输入/输出系统(BIOS)可以典型地存储在ROM中,该基本输入/输出系统(BIOS)包含有助于诸如在启动期间在计算机系统1600内的元件之间传递信息的基本例程。RAM通常包含当前由处理子系统1604操作和执行的数据和/或程序模块。在一些实现中,系统存储器1610可以包括多种不同类型的存储器,诸如静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)等。

[0293] 作为示例而非限制,如图16中所示,系统存储器1610可以加载正在被执行的可以包括各种应用(诸如Web浏览器、中间层应用、关系型数据库管理系统(RDBMS)等)的应用程序1612、程序数据1614和操作系统1616。作为示例,操作系统1616可以包括各种版本的Microsoft **Windows®**、Apple **Macintosh®**和/或Linux操作系统、各种商用**UNIX®**或类UNIX操作系统(包括但不限于各种GNU/Linux操作系统、Google **Chrome®** OS等)和/或移动操作系统,诸如**iOS®**、**Windows®** Phone、**Android®** OS、**BlackBerry®** OS、**Palm®** OS操作系统以及其它操作系统。

[0294] 计算机可读存储介质1622可以存储提供一些实施例的功能的编程和数据构造。计算机可读存储介质1622可以为计算机系统1600提供计算机可读指令、数据结构、程序模块和其它数据的存储。当由处理子系统1604执行时,提供上述功能的软件(程序、代码模块、指令)可以存储在存储子系统1618中。作为示例,计算机可读存储介质1622可以包括非易失性存储器,诸如硬盘驱动器、磁盘驱动器、诸如CD ROM、DVD、**Blu-Ray®**(蓝光)盘或其它光学介质的光盘驱动器。计算机可读存储介质1622可以包括但不限于,**Zip®**驱动器、闪存存储器卡、通用串行总线(USB)闪存驱动器、安全数字(SD)卡、DVD盘、数字视频带等。计算机可读存储介质1622也可以包括基于非易失性存储器的固态驱动器(SSD)(诸如基于闪存存储器的SSD、企业闪存驱动器、固态ROM等)、基于易失性存储器的SSD(诸如基于固态RAM、动态RAM、静态RAM、基于DRAM的SSD、磁阻RAM(MRAM)SSD),以及使用基于DRAM和基于闪存存储器的SSD的组合的混合SSD。

[0295] 在某些实施例中,存储子系统1618还可以包括计算机可读存储介质读取器1620,其还可以连接到计算机可读存储介质1622。读取器1620可以接收并且被配置为从诸如盘、闪存驱动器等的存储器设备读取数据。

[0296] 在某些实施例中,计算机系统1600可以支持虚拟化技术,包括但不限于处理和存储器资源的虚拟化。例如,计算机系统1600可以提供用于执行一个或多个虚拟机的支持。在

某些实施例中,计算机系统1600可以执行诸如促进虚拟机的配置和管理的管理程序之类的程序。可以为每个虚拟机分配存储器、计算(例如,处理器、核)、I/O和联网资源。每个虚拟机通常独立于其它虚拟机运行。虚拟机通常运行其自己的操作系统,该操作系统可以与由计算机系统1600执行的其它虚拟机所执行的操作系统相同或不同。因此,计算机系统1600可以潜在地同时运行多个操作系统。

[0297] 通信子系统1624提供到其它计算机系统和网络的界面。通信子系统1624用作用于从其它系统接收数据以及从计算机系统1600向其它系统传输数据的界面。例如,通信子系统1624可以使得计算机系统1600能够经由互联网建立到一个或多个客户端设备的通信信道,以用于从客户端设备接收信息以及向客户端设备发送信息。

[0298] 通信子系统1624可以支持有线和/或无线通信协议两者。例如,在某些实施例中,通信子系统1624可以包括用于(例如,使用蜂窝电话技术、高级数据网络技术(诸如3G、4G或EDGE(全球演进的增强数据速率)、WiFi(IEEE 802.XX族标准)、或其它移动通信技术、或其任意组合)接入无线语音和/或数据网络的射频(RF)收发器组件、全球定位系统(GPS)接收器组件和/或其它组件。在一些实施例中,作为无线接口的附加或替代,通信子系统1624可以提供有线网络连接(例如,以太网)。

[0299] 通信子系统1624可以以各种形式接收和传输数据。例如,在一些实施例中,除了其它形式之外,通信子系统1624还可以以结构化和/或非结构化的数据馈送1626、事件流1628、事件更新1630等形式接收输入通信。例如,通信子系统1624可以被配置为实时地从社交媒体网络的用户和/或诸如**Twitter®**馈送、**Facebook®**更新、诸如丰富站点摘要(RSS)馈送的web馈送的其它通信服务接收(或发送)数据馈送1626,和/或来自一个或多个第三方信息源的实时更新。

[0300] 在某些实施例中,通信子系统1624可以被配置为以连续数据流的形式接收本质上可能是连续的或无界的没有明确结束的数据,其中连续数据流可以包括实时事件的事件流1628和/或事件更新1630。生成连续数据的应用的示例可以包括例如传感器数据应用、金融报价机、网络性能测量工具(例如网络监视和流量管理应用)、点击流分析工具、汽车流量监视等。

[0301] 通信子系统1624也可以被配置为将数据从计算机系统1600传送到其它计算机系统或网络。数据可以以各种不同的形式(诸如结构化和/或非结构化数据馈送1626、事件流1628、事件更新1630等)传送给一个或多个数据库,该一个或多个数据库可以与耦合到计算机系统1600的一个或多个流传输数据源进行通信。

[0302] 计算机系统1600可以是各种类型中的一种,包括手持便携式设备(例如,**iPhone®**蜂窝电话、**iPad®**计算平板、PDA)、可穿戴设备(例如,Google**Glass®**头戴式显示器)、个人计算机、工作站、大型机、信息站、服务器机架或任何其它数据处理系统。由于计算机和网络不断变化的性质,对图16中绘出的计算机系统1600的描述旨在仅仅作为具体示例。具有比图16中所绘出的系统更多或更少组件的许多其它配置是可能的。基于本文所提供的公开内容和教导,本领域普通技术人员将理解实现各种实施例的其它方式和/或方法。

[0303] 虽然已经描述了特定的实施例,但是各种修改、变更、替代构造以及等同物都是可能的。例如,虽然已经使用SQL查询描述了某些示例,但这并不意味着限制。在替代实施例

中,本文描述的教导还可以应用于除SQL查询以外的查询。实施例不限于在某些特定数据处理环境内的操作,而是可以在多个数据处理环境内自由操作。此外,虽然已经使用一系列特定的事务和步骤描述了某些实施例,但是对于本领域技术人员来说明显的是,这并不旨在进行限制。虽然一些流程图将操作描述为顺序处理,但是许多操作可以并行或同时执行。此外,操作的次序可以被重新布置。处理可能具有图中未包括的其它步骤。上述实施例的各种特征和方面可以被单独使用或联合使用。

[0304] 另外,虽然已经使用硬件和软件的特定组合描述了某些实施例,但是应该认识到的是,硬件和软件的其它组合也是可能的。某些实施例可以仅用硬件或仅用软件或其组合来实现。本文描述的各种处理可以以任何组合在相同的处理器或不同的处理器上实现。

[0305] 在将设备、系统、组件或模块描述为被配置为执行某些操作或功能的情况下,这样的配置可以通过以下方式来实现,例如,通过设计电子电路来执行操作、通过对可编程电子电路(诸如微处理器)进行编程来执行操作,诸如通过执行计算机指令或代码,或处理器或核心被编程为执行存储在非暂态存储介质上的代码或指令,或其任意组合来执行操作。进程可以使用各种技术进行通信,包括但不限于用于进程间通信的常规技术,并且不同对的进程可以使用不同的技术,或者同一对进程可以在不同时间使用不同的技术。

[0306] 在本公开中给出了具体细节以提供对实施例的透彻理解。但是,可以在没有这些具体细节的情况下实践实施例。例如,已经示出了众所周知的电路、处理、算法、结构和技术,而没有不必要的细节,以避免使实施例模糊。本描述仅提供示例实施例,并且不旨在限制其它实施例的范围、适用性或配置。而是,实施例的先前描述将为本领域技术人员提供用于实现各种实施例的使能描述。可以对元件的功能和布置进行各种改变。

[0307] 因此,说明书和附图应被认为是说明性的而不是限制性的。但是,将明显的是,在不脱离权利要求书所阐述的更广泛的精神和范围的情况下,可以对其进行添加、减少、删除以及其它修改和改变。因此,虽然已经描述了具体的实施例,但是这些实施例并不旨在进行限制。各种修改和等同物在所附权利要求的范围内。

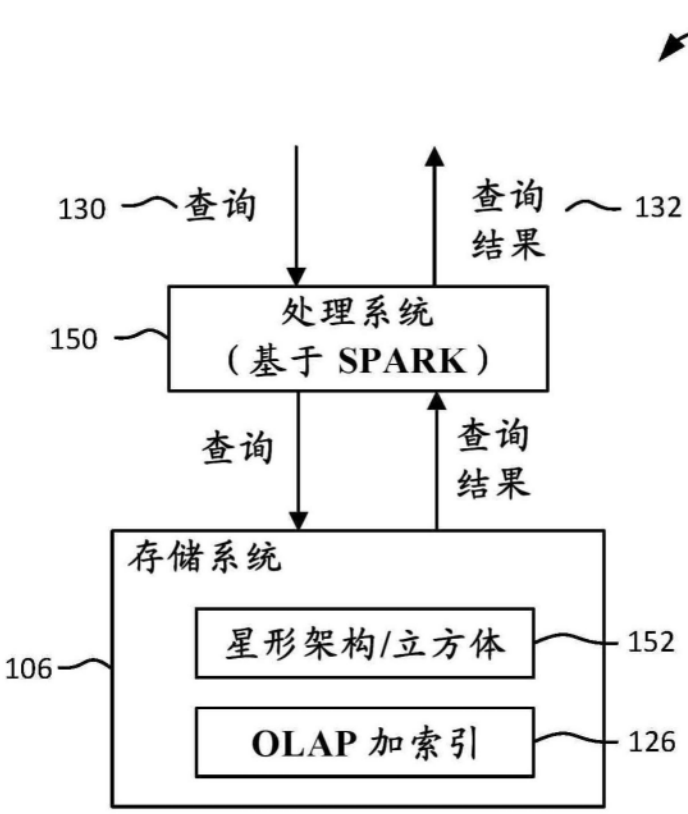

100
130
查询
查询
结果
132
150
处理系统
（基于 SPARK）
查询
查询
结果
存储系统
106
星形架构/立方体
152
OLAP 加索引
126

图1A

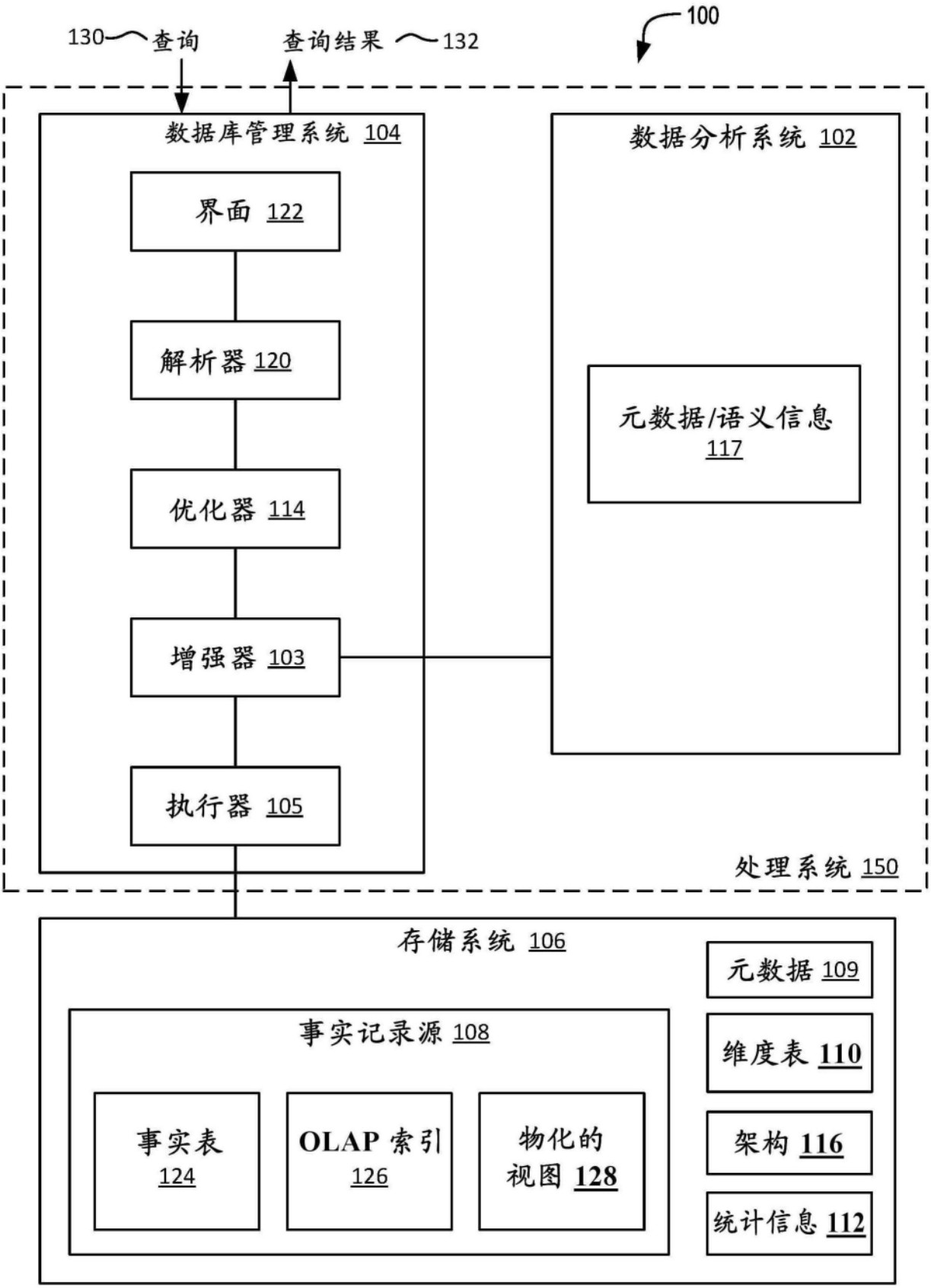

130
查询
查询结果
132
100
数据库管理系统 104
界面 122
解析器 120
优化器 114
增强器 103
执行器 105
数据分析系统 102
元数据/语义信息
117
处理系统 150
存储系统 106
元数据 109
事实记录源 108
维度表 110
事实表
124
OLAP 索引
126
物化的
视图 128
架构 116
统计信息 112

图1B

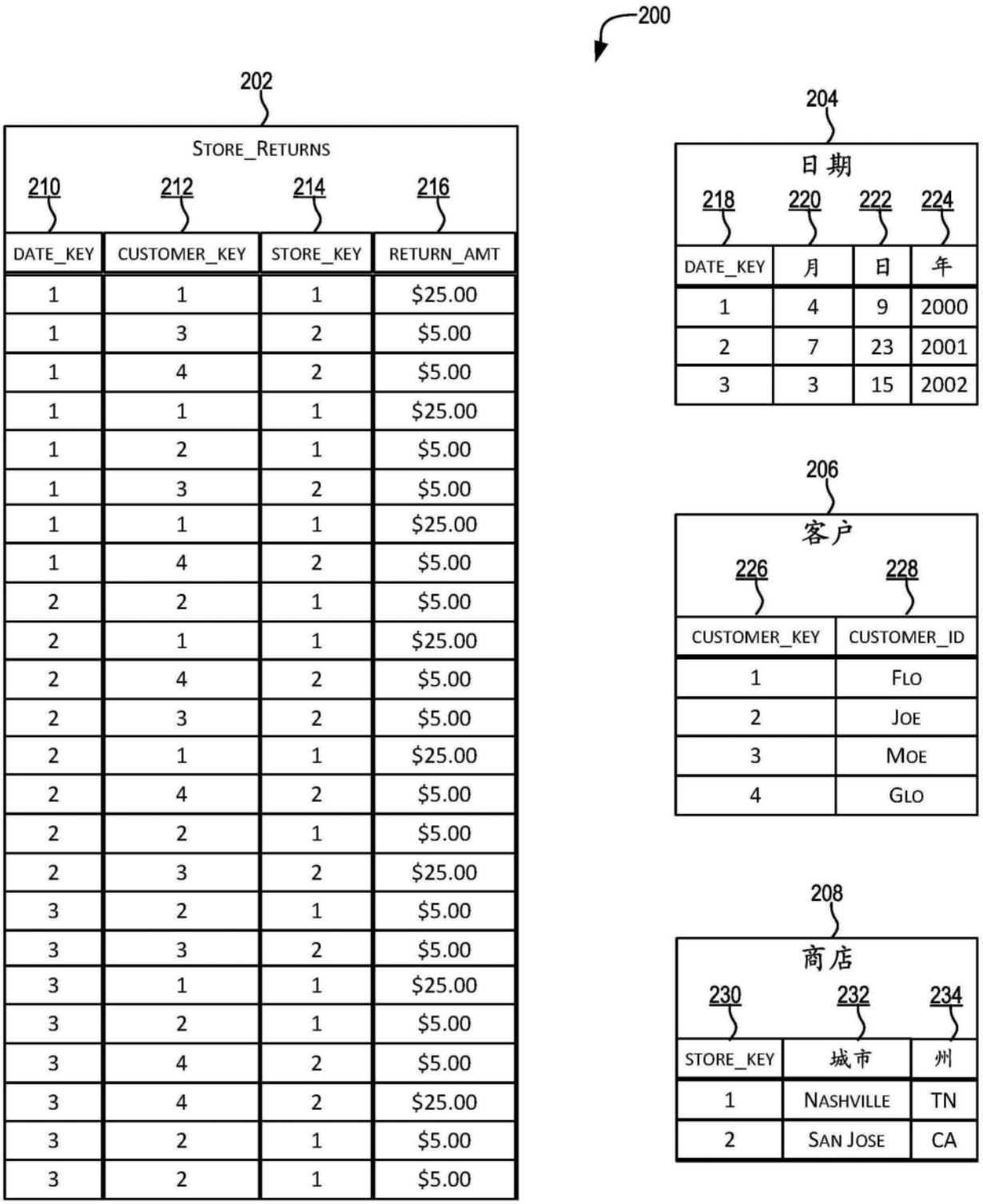

200
202
STORE_RETURNS
210 212 214 216
DATE_KEY CUSTOMER_KEY STORE_KEY RETURN_AMT
1 1 1 $25.00
1 3 2 $5.00
1 4 2 $5.00
1 1 1 $25.00
1 2 1 $5.00
1 3 2 $5.00
1 1 1 $25.00
1 4 2 $5.00
2 2 1 $5.00
2 1 1 $25.00
2 4 2 $5.00
2 3 2 $5.00
2 1 1 $25.00
2 4 2 $5.00
2 2 1 $5.00
2 3 2 $25.00
3 2 1 $5.00
3 3 2 $5.00
3 1 1 $25.00
3 2 1 $5.00
3 4 2 $5.00
3 4 2 $25.00
3 2 1 $5.00
3 2 1 $5.00
204
日期
218 220 222 224
DATE_KEY 月 日 年
1 4 9 2000
2 7 23 2001
3 3 15 2002
206
客户
226 228
CUSTOMER_KEY CUSTOMER_ID
1 FLO
2 JOE
3 MOE
4 GLO
208
商店
230 232 234
STORE_KEY 城市 州
1 NASHVILLE TN
2 SAN JOSE CA

图2

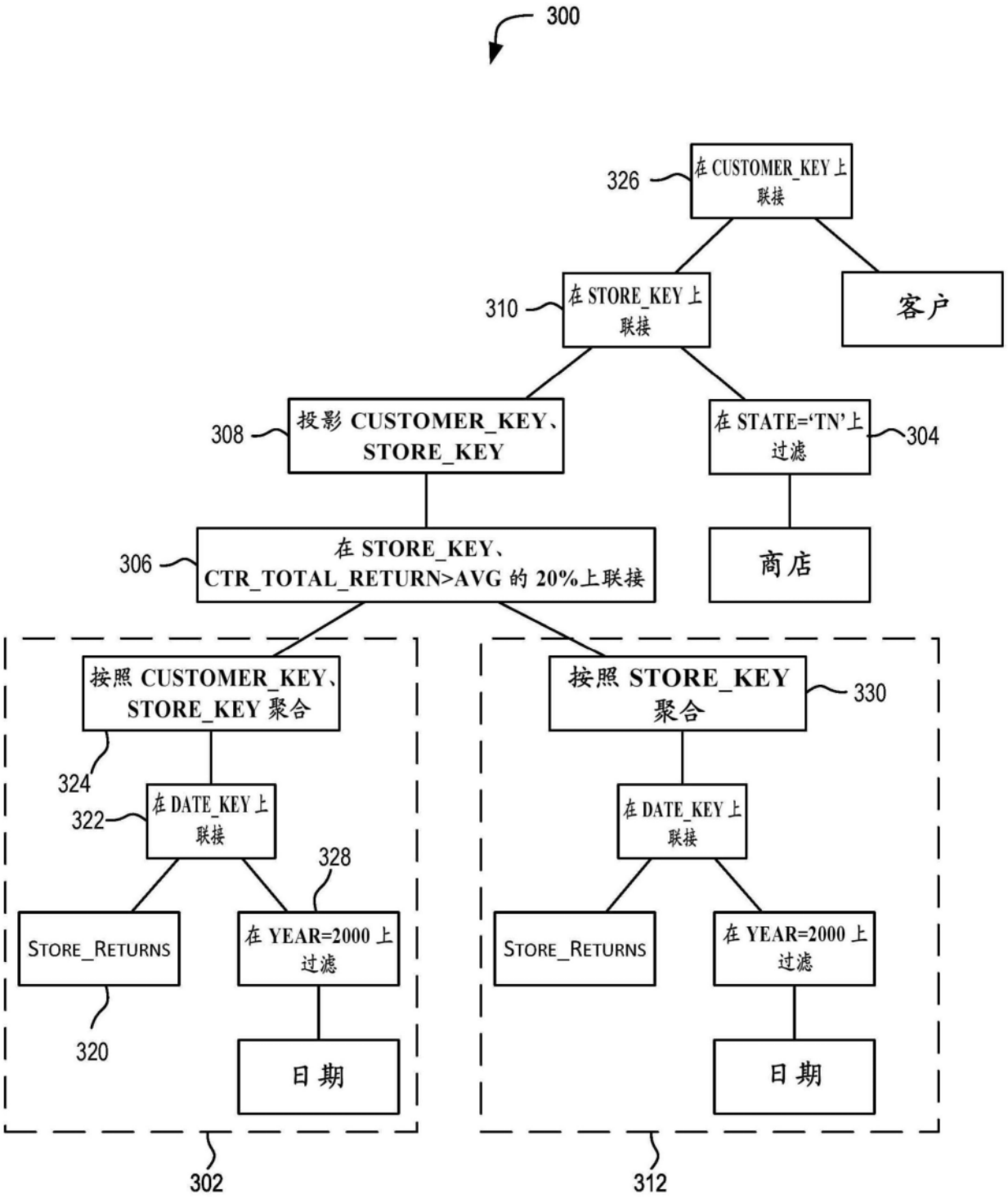

300
326
在 CUSTOMER_KEY 上联接
310
在 STORE_KEY 上联接
客户
308
投影 CUSTOMER_KEY、STORE_KEY
在 STATE='TN'上过滤
304
306
在 STORE_KEY、CTR_TOTAL_RETURN>AVG 的 20%上联接
商店
按照 CUSTOMER_KEY、STORE_KEY 聚合
324
322
在 DATE_KEY 上联接
328
STORE_RETURNS
在 YEAR=2000 上过滤
320
日 期
302
按照 STORE_KEY 聚合
330
在 DATE_KEY 上联接
STORE_RETURNS
在 YEAR=2000 上过滤
日 期
312

图3

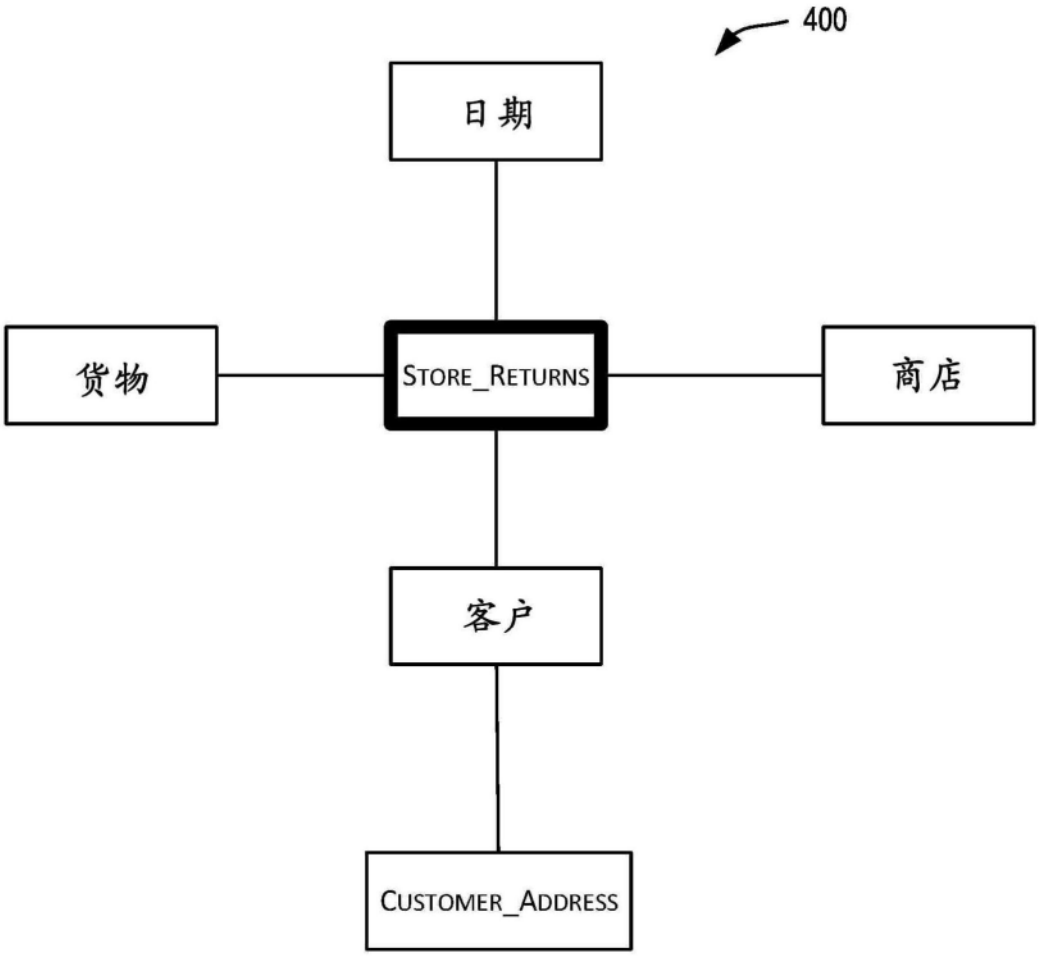



图4

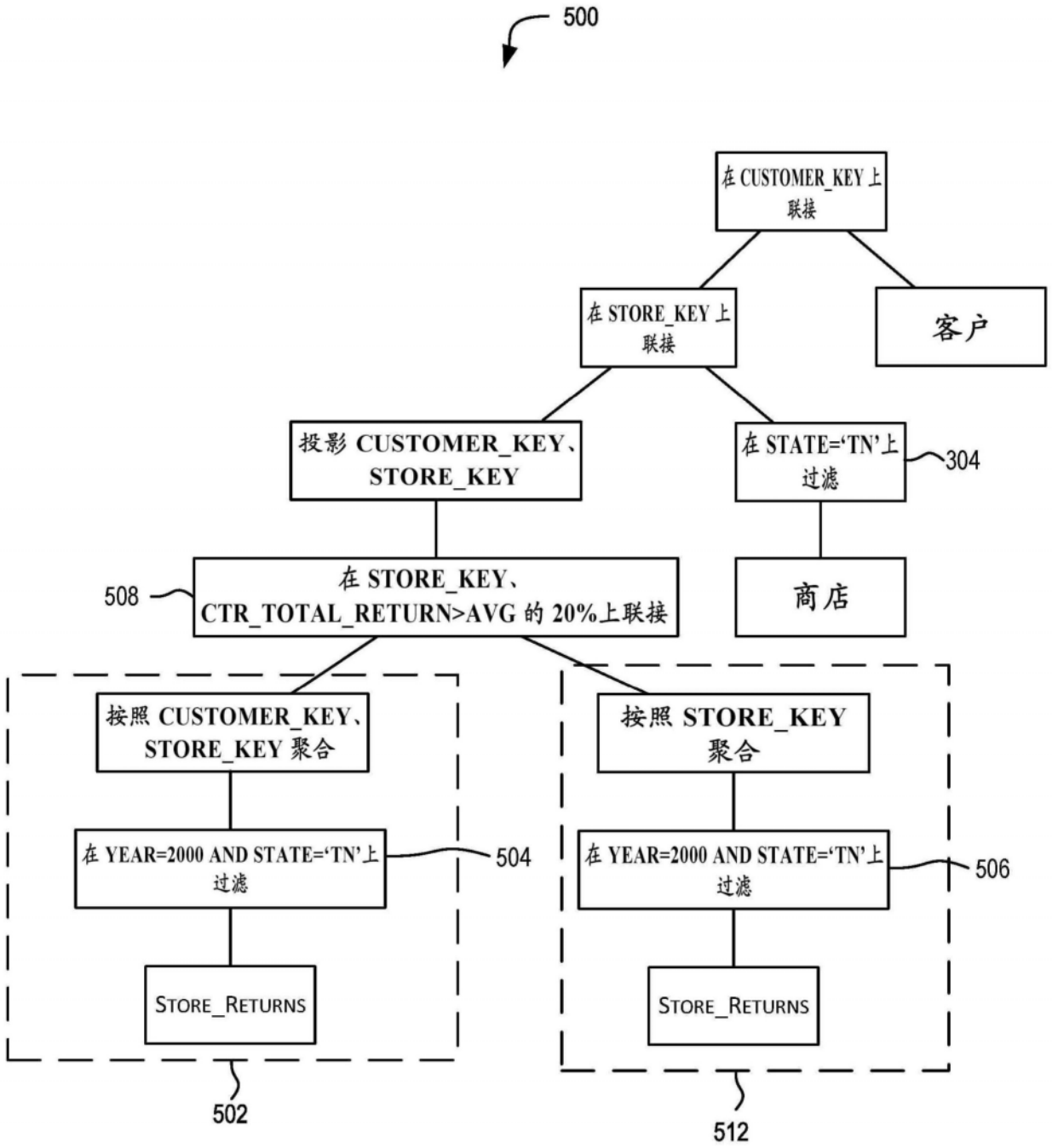

500
在 CUSTOMER_KEY 上联接
在 STORE_KEY 上联接
客户
投影 CUSTOMER_KEY、STORE_KEY
在 STATE='TN'上过滤
304
508
在 STORE_KEY、CTR_TOTAL_RETURN>AVG 的 20%上联接
商店
按照 CUSTOMER_KEY、STORE_KEY 聚合
按照 STORE_KEY 聚合
在 YEAR=2000 AND STATE='TN'上过滤
504
在 YEAR=2000 AND STATE='TN'上过滤
506
STORE_RETURNS
STORE_RETURNS
502
512

图5

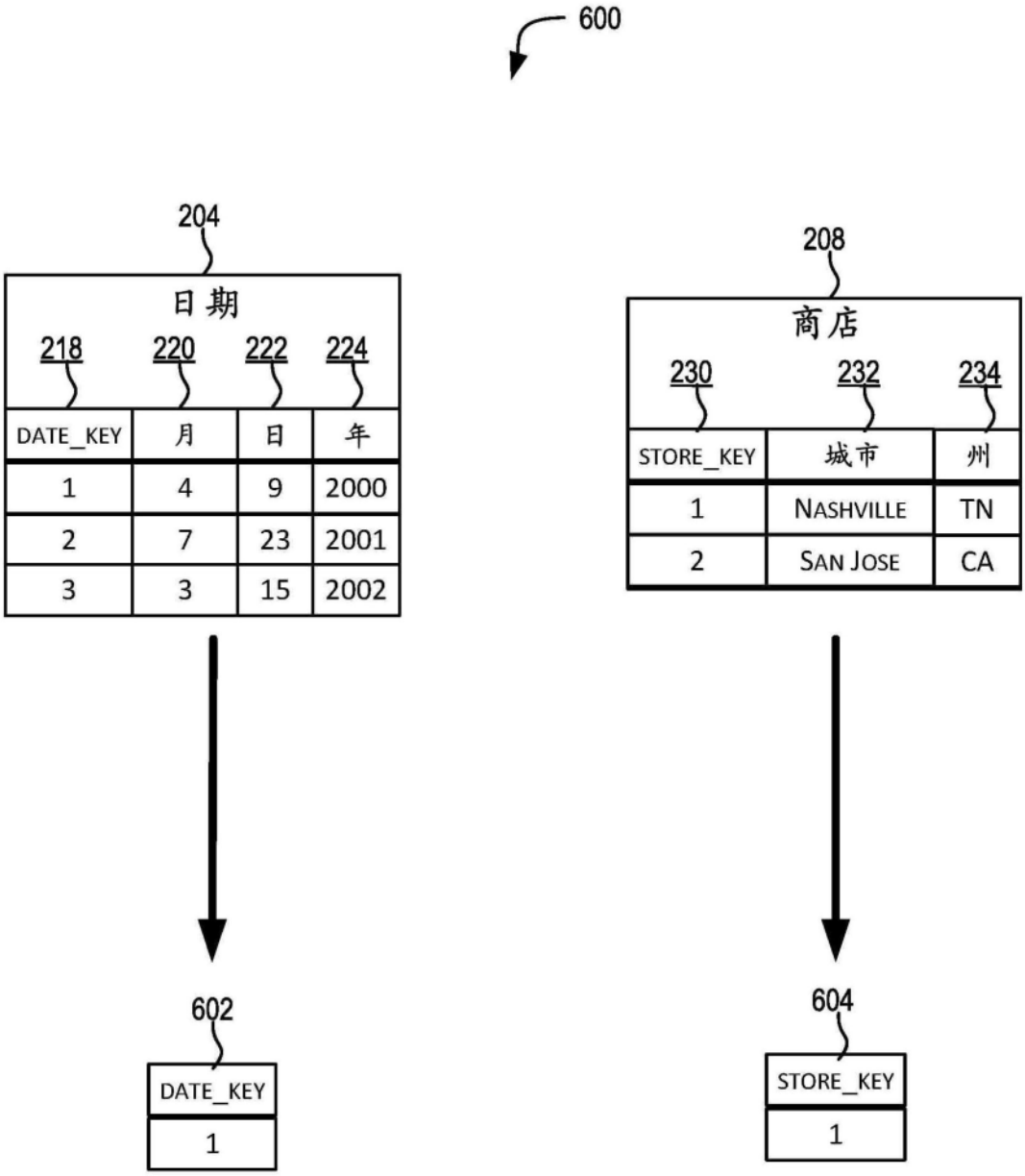

600
204
日期
218
220
222
224
DATE_KEY
月
日
年
1
4
9
2000
2
7
23
2001
3
3
15
2002
208
商店
230
232
234
STORE_KEY
城市
州
1
NASHVILLE
TN
2
SAN JOSE
CA
602
DATE_KEY
1
604
STORE_KEY
1

图6

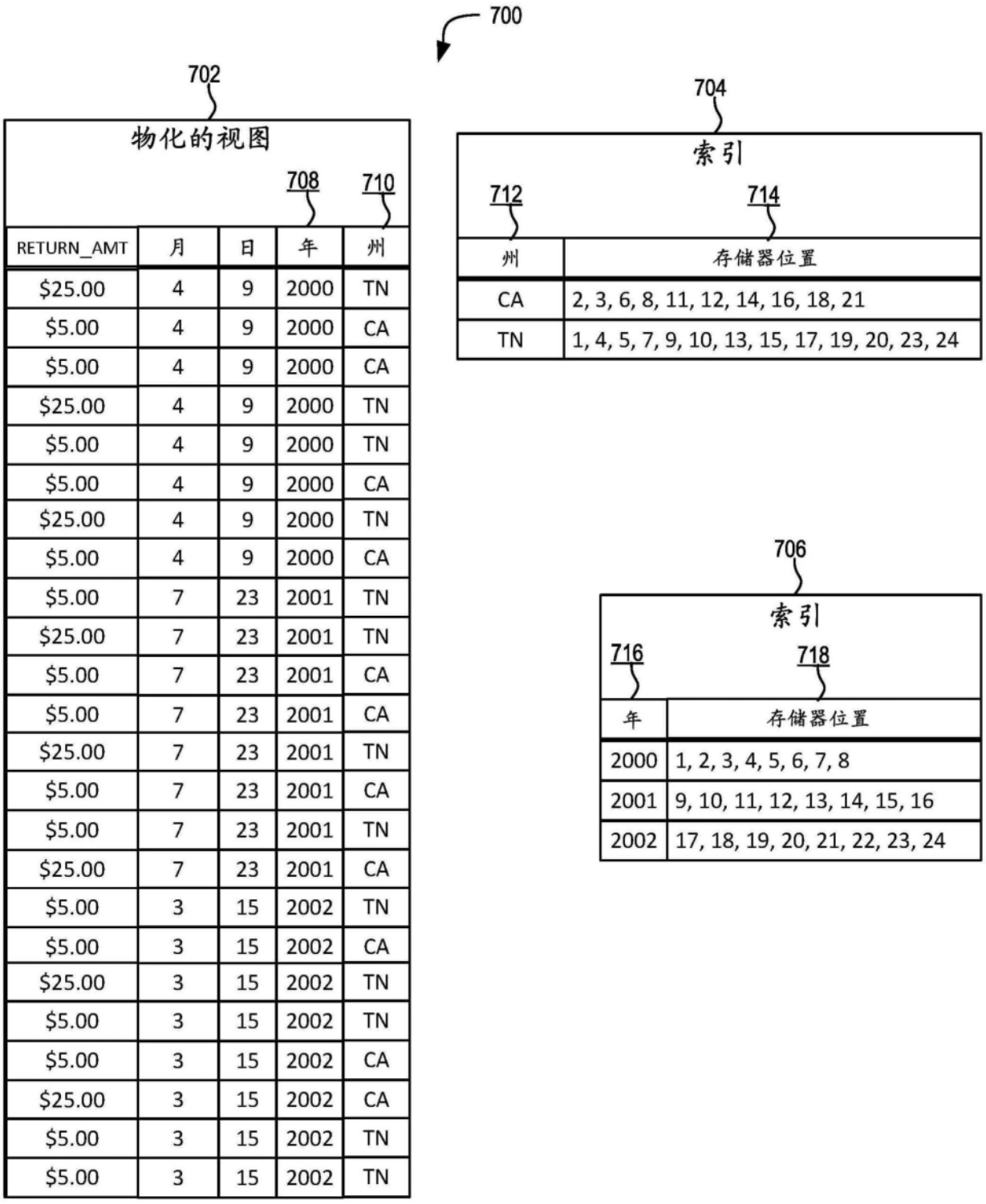

700
702
物化的视图
708
710
RETURN_AMT
月
日
年
州
$25.00 4 9 2000 TN
$5.00 4 9 2000 CA
$5.00 4 9 2000 CA
$25.00 4 9 2000 TN
$5.00 4 9 2000 TN
$5.00 4 9 2000 CA
$25.00 4 9 2000 TN
$5.00 4 9 2000 CA
$5.00 7 23 2001 TN
$25.00 7 23 2001 TN
$5.00 7 23 2001 CA
$5.00 7 23 2001 CA
$25.00 7 23 2001 TN
$5.00 7 23 2001 CA
$5.00 7 23 2001 TN
$25.00 7 23 2001 CA
$5.00 3 15 2002 TN
$5.00 3 15 2002 CA
$25.00 3 15 2002 TN
$5.00 3 15 2002 TN
$5.00 3 15 2002 CA
$25.00 3 15 2002 CA
$5.00 3 15 2002 TN
$5.00 3 15 2002 TN
704
索引
712
714
州
存储器位置
CA 2, 3, 6, 8, 11, 12, 14, 16, 18, 21
TN 1, 4, 5, 7, 9, 10, 13, 15, 17, 19, 20, 23, 24
706
索引
716
718
年
存储器位置
2000 1, 2, 3, 4, 5, 6, 7, 8
2001 9, 10, 11, 12, 13, 14, 15, 16
2002 17, 18, 19, 20, 21, 22, 23, 24

图7

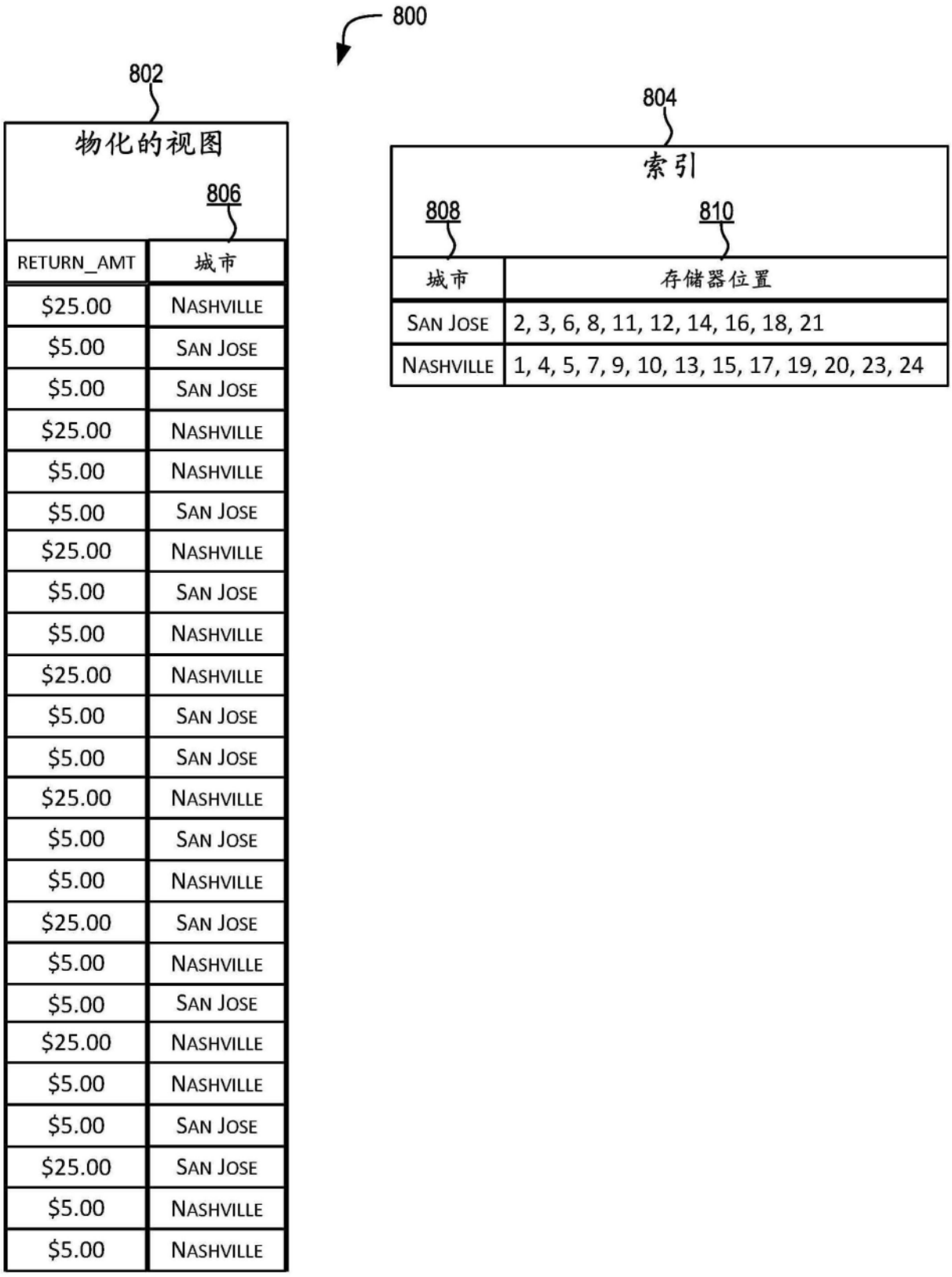

800
802
物化的视图
806
RETURN_AMT
城市
$25.00 NASHVILLE
$5.00 SAN JOSE
$5.00 SAN JOSE
$25.00 NASHVILLE
$5.00 NASHVILLE
$5.00 SAN JOSE
$25.00 NASHVILLE
$5.00 SAN JOSE
$5.00 NASHVILLE
$25.00 NASHVILLE
$5.00 SAN JOSE
$5.00 SAN JOSE
$25.00 NASHVILLE
$5.00 SAN JOSE
$5.00 NASHVILLE
$25.00 SAN JOSE
$5.00 NASHVILLE
$5.00 SAN JOSE
$25.00 NASHVILLE
$5.00 NASHVILLE
$5.00 SAN JOSE
$25.00 SAN JOSE
$5.00 NASHVILLE
$5.00 NASHVILLE
804
索引
808
810
城市
存储器位置
SAN JOSE 2, 3, 6, 8, 11, 12, 14, 16, 18, 21
NASHVILLE 1, 4, 5, 7, 9, 10, 13, 15, 17, 19, 20, 23, 24

图8A

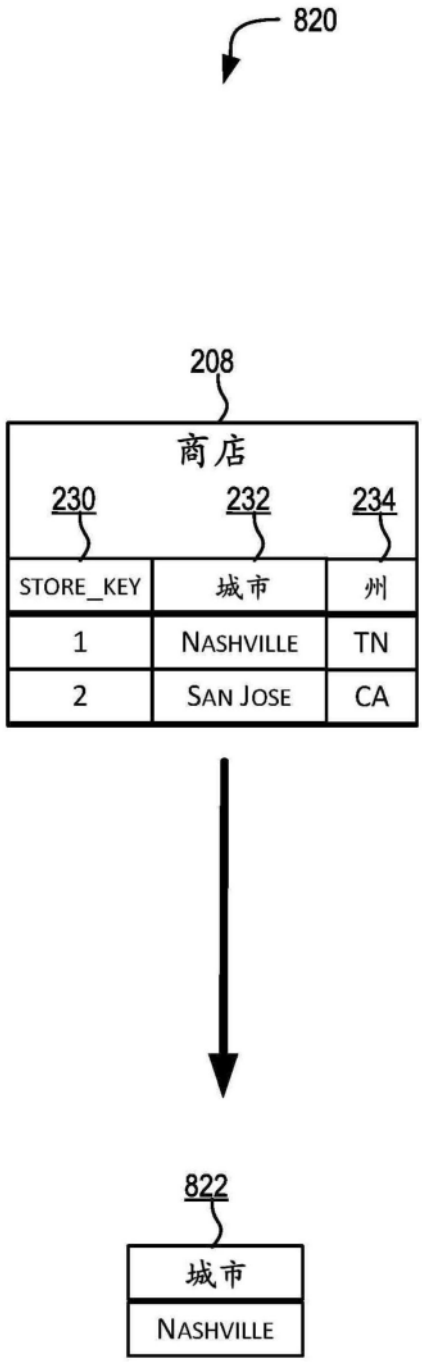
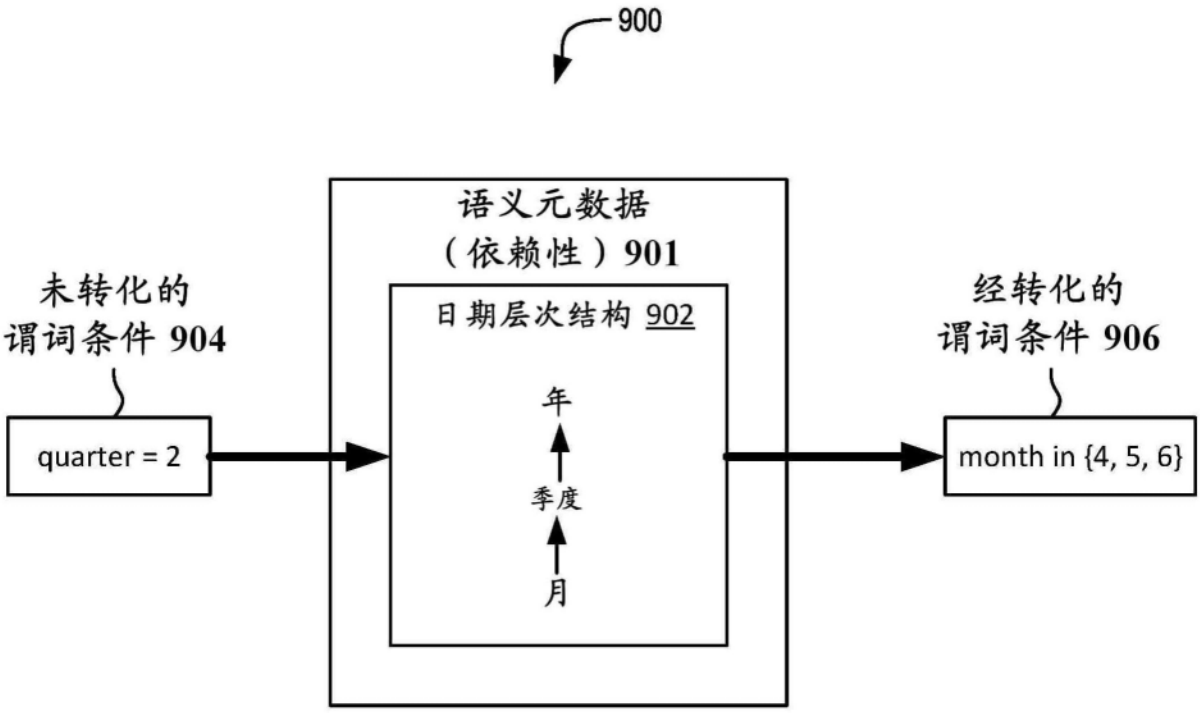

820
208
商店
230
232
234
STORE_KEY
城市
州
1
NASHVILLE
TN
2
SAN JOSE
CA
822
城市
NASHVILLE

图8B


900
语义元数据
（依赖性）901
日期层次结构 902
年
季度
月
未转化的
谓词条件 904
quarter = 2
经转化的
谓词条件 906
month in {4, 5, 6}

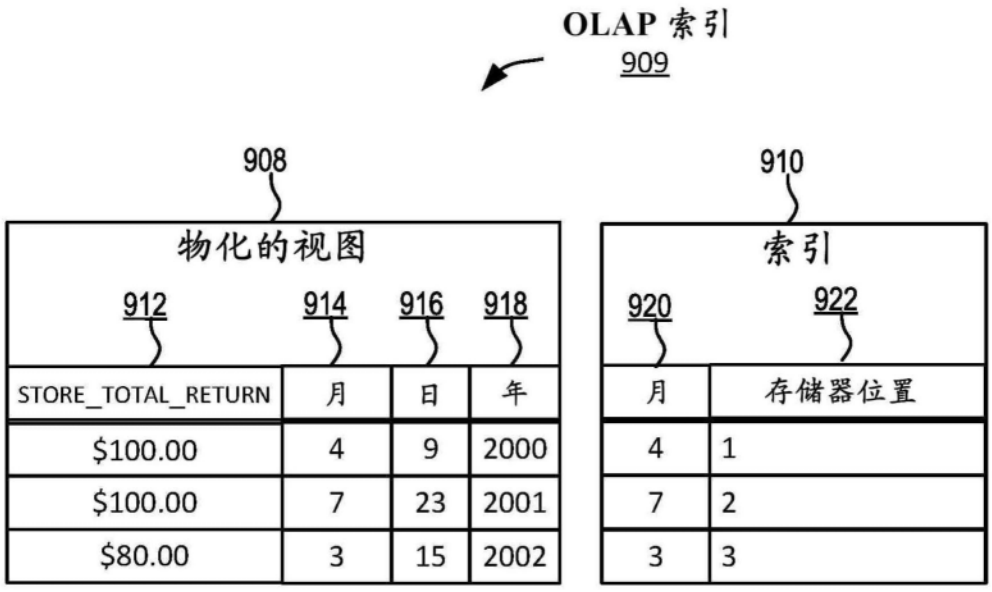

OLAP 索引
909
908
物化的视图
912
914
916
918
STORE_TOTAL_RETURN
月
日
年
$100.00
4
9
2000
$100.00
7
23
2001
$80.00
3
15
2002
910
索引
920
922
月
存储器位置
4
1
7
2
3
3

图9

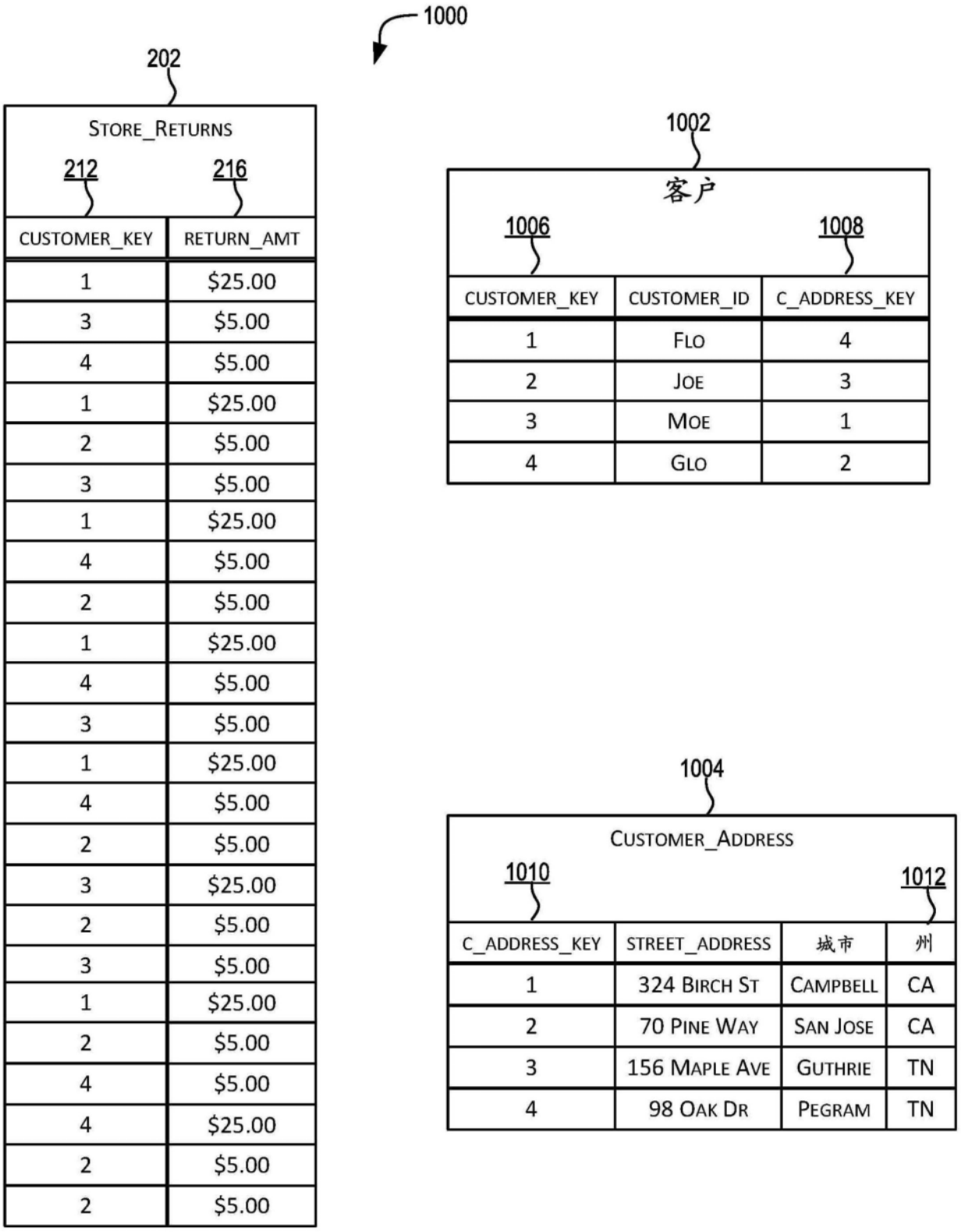
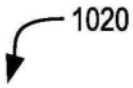

1000
202
STORE_RETURNS
212
216
CUSTOMER_KEY
RETURN_AMT
1 $25.00
3 $5.00
4 $5.00
1 $25.00
2 $5.00
3 $5.00
1 $25.00
4 $5.00
2 $5.00
1 $25.00
4 $5.00
3 $5.00
1 $25.00
4 $5.00
2 $5.00
3 $25.00
2 $5.00
3 $5.00
1 $25.00
2 $5.00
4 $5.00
4 $25.00
2 $5.00
2 $5.00
1002
客户
1006
1008
CUSTOMER_KEY
CUSTOMER_ID
C_ADDRESS_KEY
1 FLO 4
2 JOE 3
3 MOE 1
4 GLO 2
1004
CUSTOMER_ADDRESS
1010
1012
C_ADDRESS_KEY
STREET_ADDRESS
城市
州
1 324 BIRCH ST CAMPBELL CA
2 70 PINE WAY SAN JOSE CA
3 156 MAPLE AVE GUTHRIE TN
4 98 OAK DR PEGRAM TN

图10A




物化的视图

1022

| CUSTOMER_KEY | RETURN_AMT | 州 |
|---|---|---|
| 1 | $25.00 | TN |
| 3 | $5.00 | CA |
| 4 | $5.00 | CA |
| 1 | $25.00 | TN |
| 2 | $5.00 | TN |
| 3 | $5.00 | CA |
| 1 | $25.00 | TN |
| 4 | $5.00 | CA |
| 2 | $5.00 | TN |
| 1 | $25.00 | TN |
| 4 | $5.00 | CA |
| 3 | $5.00 | CA |
| 1 | $25.00 | TN |
| 4 | $5.00 | CA |
| 2 | $5.00 | TN |
| 3 | $25.00 | CA |
| 2 | $5.00 | TN |
| 3 | $5.00 | CA |
| 1 | $25.00 | TN |
| 2 | $5.00 | TN |
| 4 | $5.00 | CA |
| 4 | $25.00 | CA |
| 2 | $5.00 | TN |
| 2 | $5.00 | TN |

图10B

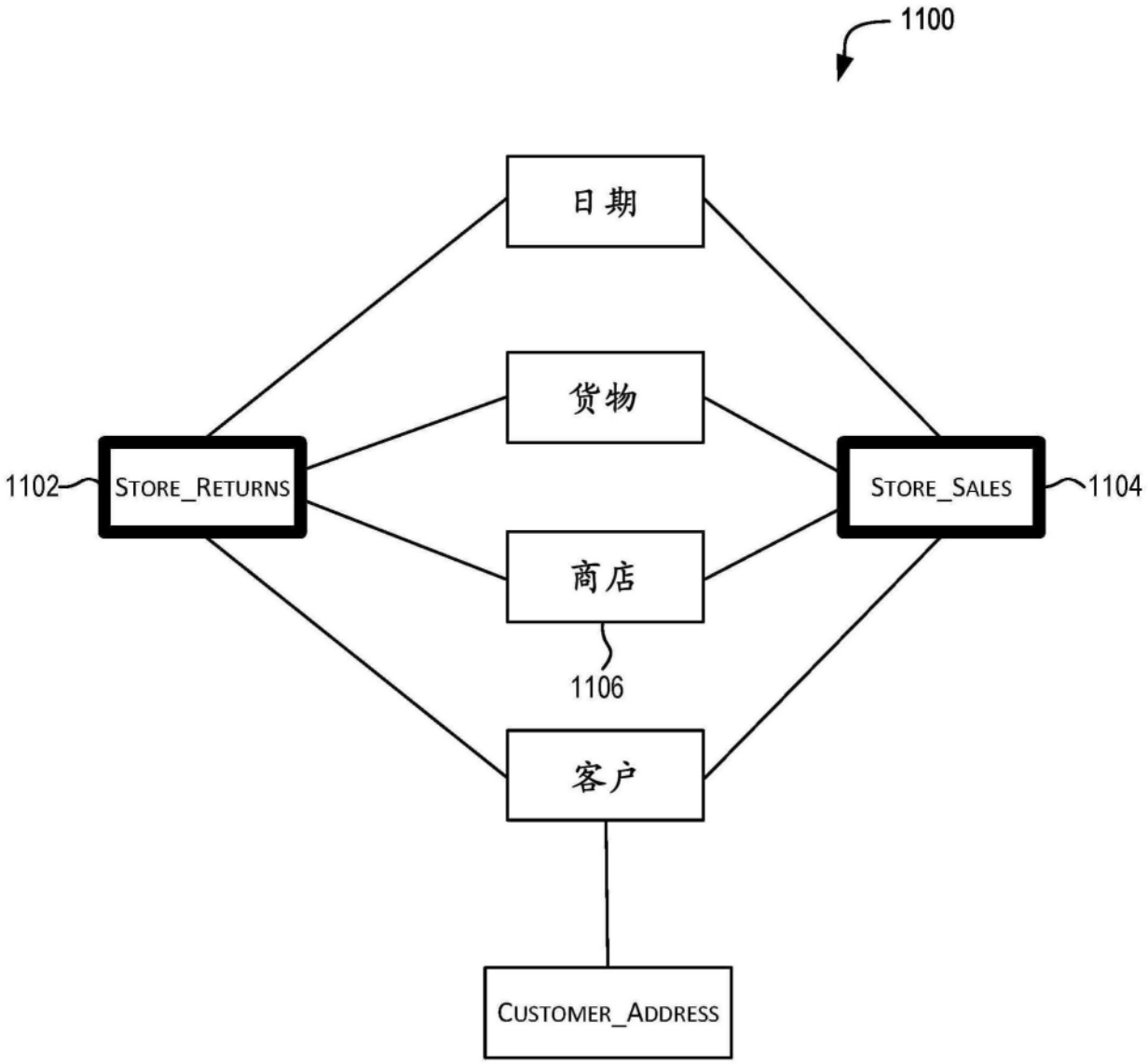

1100
日期
货物
1102
STORE_RETURNS
STORE_SALES
1104
商店
1106
客户
CUSTOMER_ADDRESS

图11

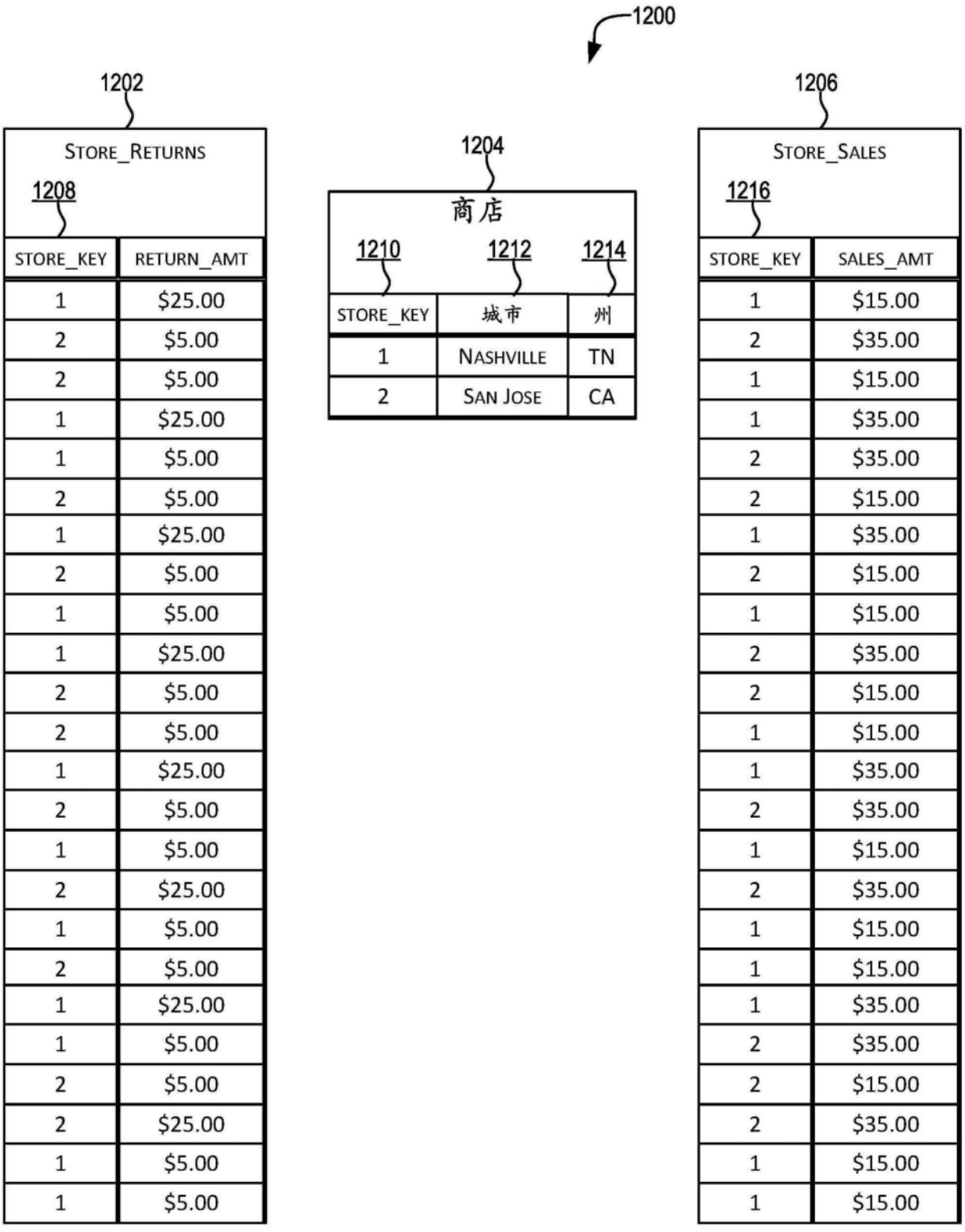

1200
1202
STORE_RETURNS
1208
STORE_KEY
RETURN_AMT
1 $25.00
2 $5.00
2 $5.00
1 $25.00
1 $5.00
2 $5.00
1 $25.00
2 $5.00
1 $5.00
1 $25.00
2 $5.00
2 $5.00
1 $25.00
2 $5.00
1 $5.00
2 $25.00
1 $5.00
2 $5.00
1 $25.00
1 $5.00
2 $5.00
2 $25.00
1 $5.00
1 $5.00
1204
商店
1210
1212
1214
STORE_KEY
城市
州
1 NASHVILLE TN
2 SAN JOSE CA
1206
STORE_SALES
1216
STORE_KEY
SALES_AMT
1 $15.00
2 $35.00
1 $15.00
1 $35.00
2 $35.00
2 $15.00
1 $35.00
2 $15.00
1 $15.00
2 $35.00
2 $15.00
1 $15.00
1 $35.00
2 $35.00
1 $15.00
2 $35.00
1 $15.00
1 $15.00
1 $35.00
2 $35.00
2 $15.00
2 $35.00
1 $15.00
1 $15.00

图12A

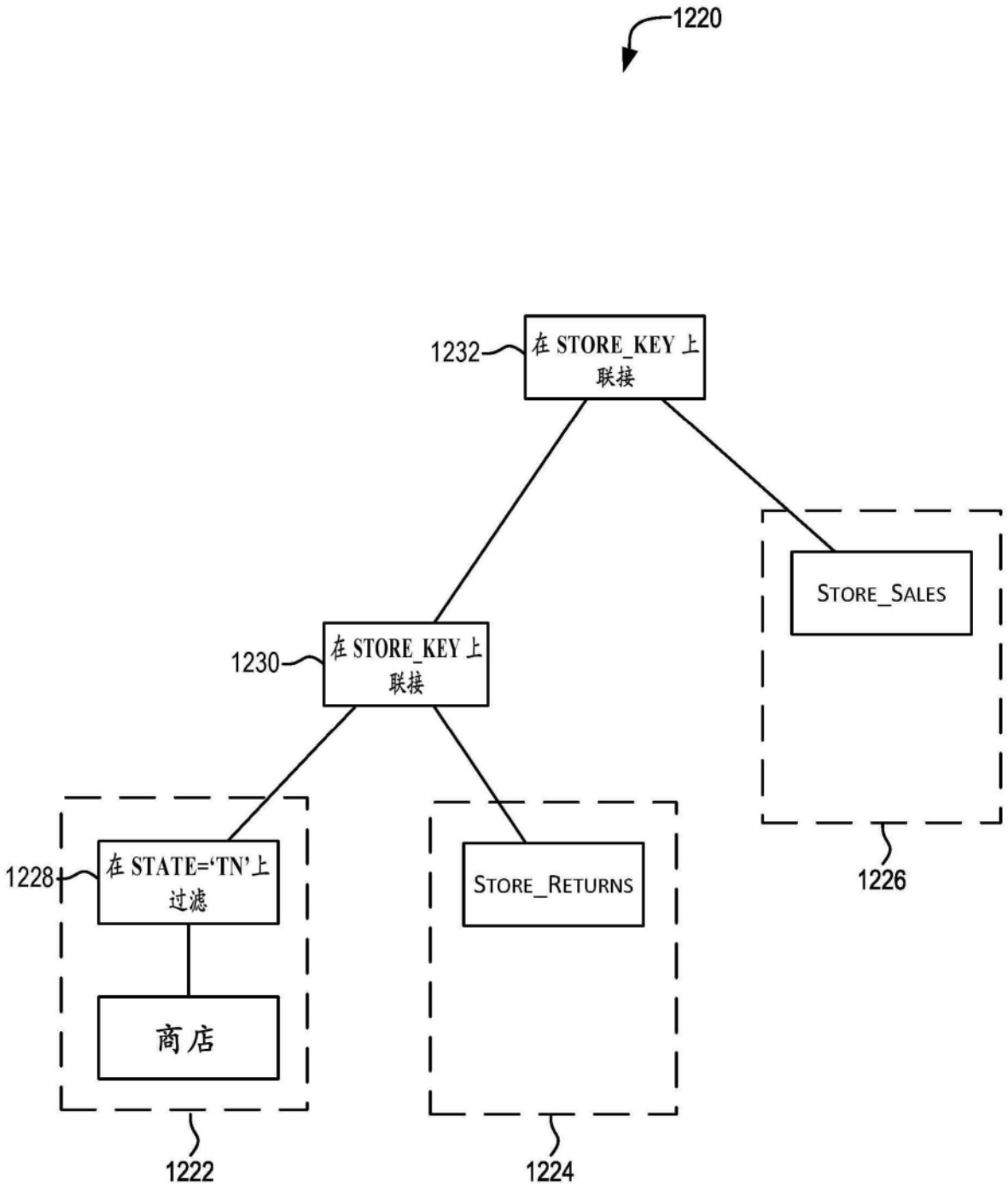

1220
1232
在 STORE_KEY 上联接
1230
在 STORE_KEY 上联接
STORE_SALES
1226
1228
在 STATE='TN'上过滤
商店
STORE_RETURNS
1222
1224

图12B

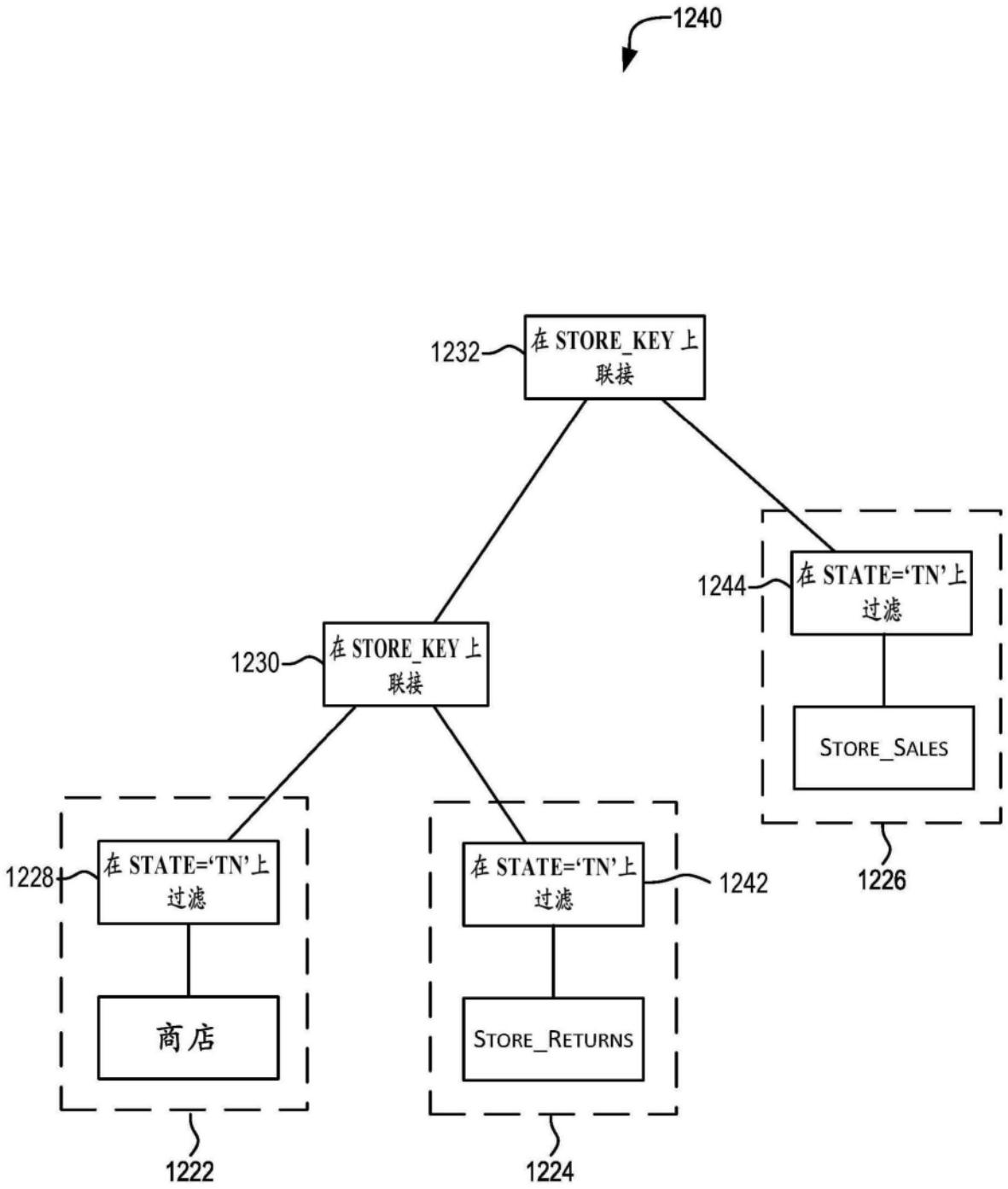

1240
1232
在 STORE_KEY 上
联接
1230
在 STORE_KEY 上
联接
1244
在 STATE='TN'上
过滤
STORE_SALES
1226
1228
在 STATE='TN'上
过滤
商店
1222
在 STATE='TN'上
过滤
1242
STORE_RETURNS
1224

图12C

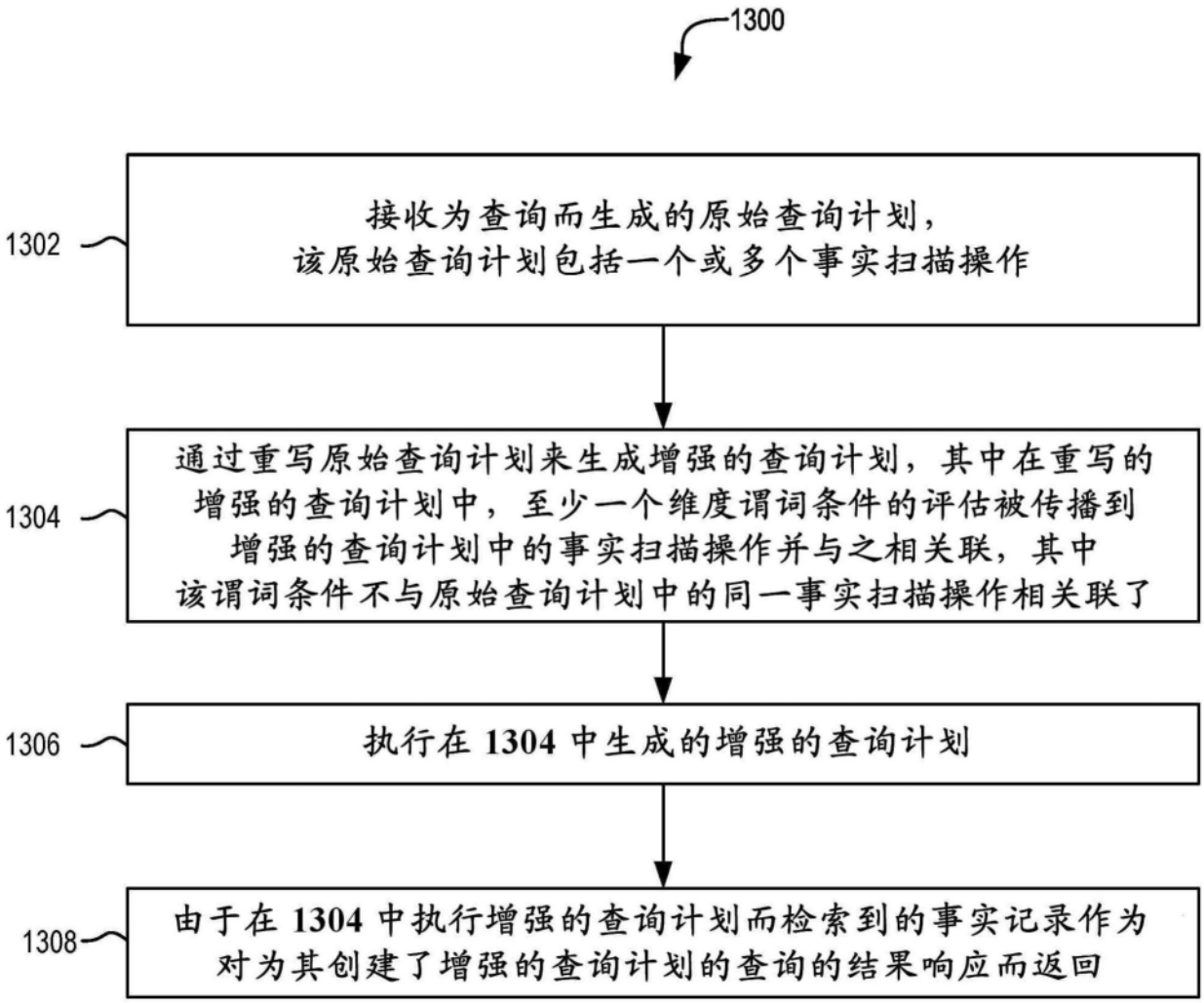

1300
1302
接收为查询而生成的原始查询计划，
该原始查询计划包括一个或多个事实扫描操作
1304
通过重写原始查询计划来生成增强的查询计划，其中在重写的
增强的查询计划中，至少一个维度谓词条件的评估被传播到
增强的查询计划中的事实扫描操作并与之相关联，其中
该谓词条件不与原始查询计划中的同一事实扫描操作相关联了
1306
执行在 1304 中生成的增强的查询计划
1308
由于在 1304 中执行增强的查询计划而检索到的事实记录作为
对为其创建了增强的查询计划的查询的结果响应而返回

图13A

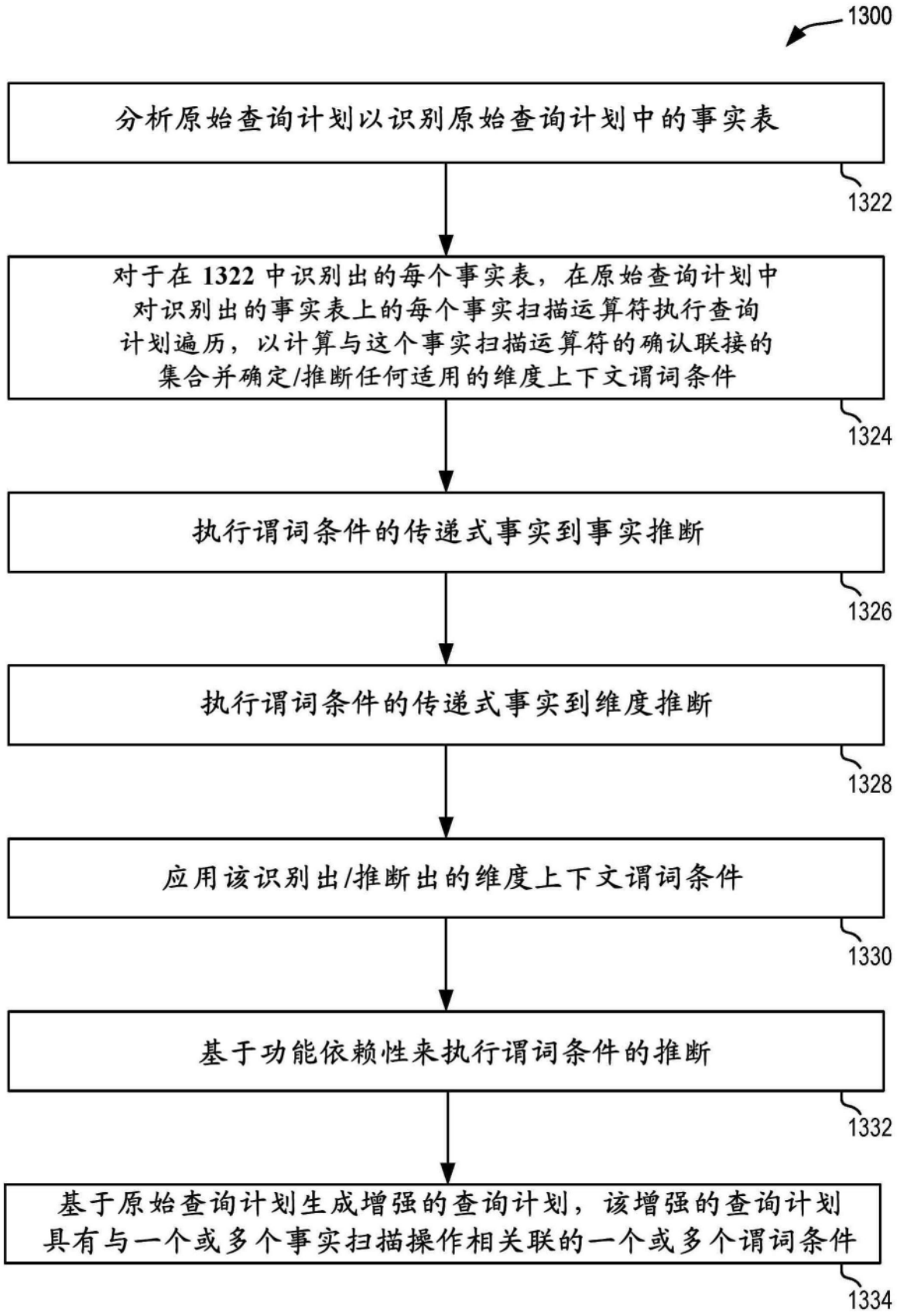

1300
分析原始查询计划以识别原始查询计划中的事实表
1322
对于在 1322 中识别出的每个事实表，在原始查询计划中对识别出的事实表上的每个事实扫描运算符执行查询计划遍历，以计算与这个事实扫描运算符的确认联接的集合并确定/推断任何适用的维度上下文谓词条件
1324
执行谓词条件的传递式事实到事实推断
1326
执行谓词条件的传递式事实到维度推断
1328
应用该识别出/推断出的维度上下文谓词条件
1330
基于功能依赖性来执行谓词条件的推断
1332
基于原始查询计划生成增强的查询计划，该增强的查询计划具有与一个或多个事实扫描操作相关联的一个或多个谓词条件
1334

图13B

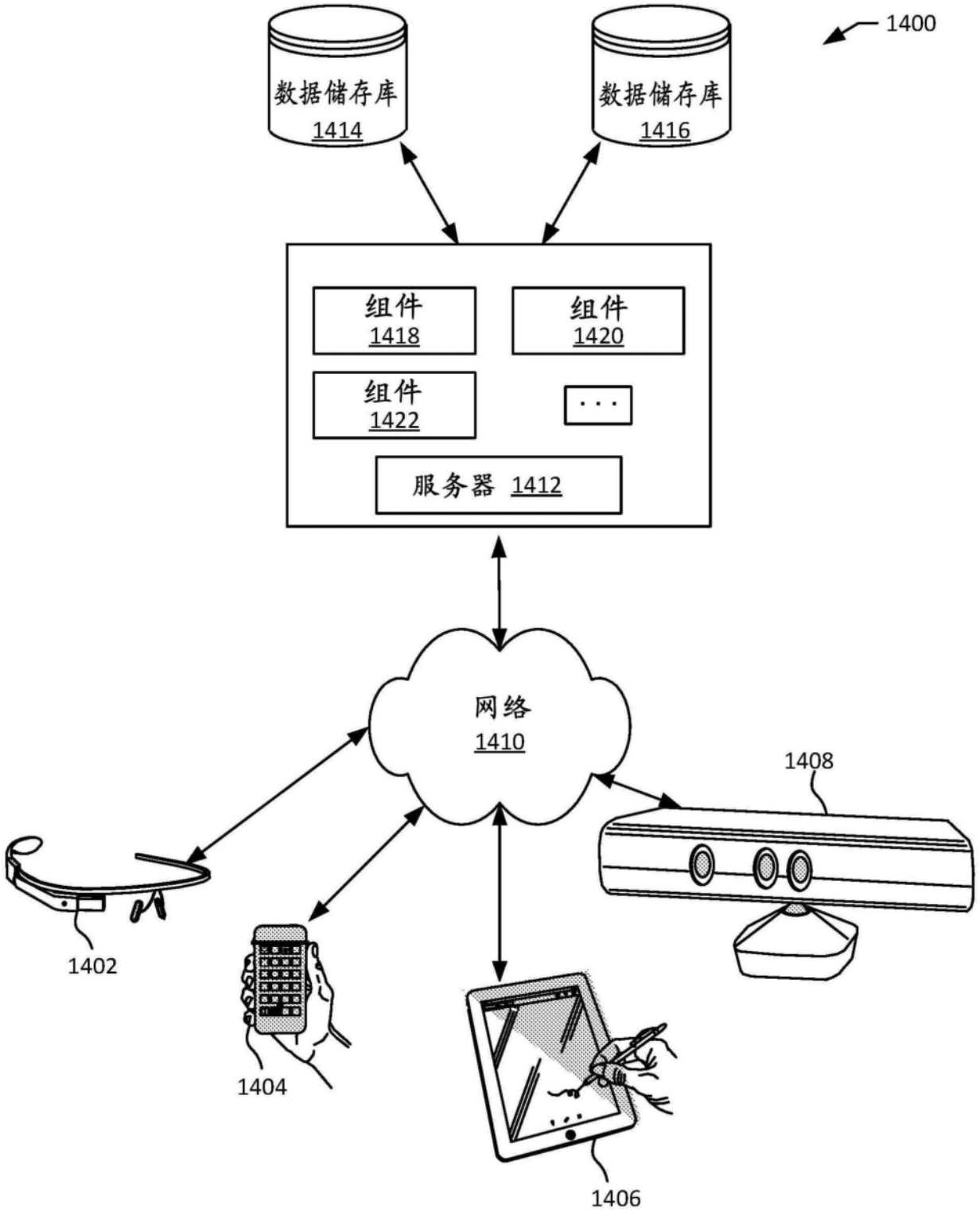

1400
数据储存库
1414
数据储存库
1416
组件
1418
组件
1420
组件
1422
· · ·
服务器 1412
网络
1410
1408
1402
1404
1406

图14

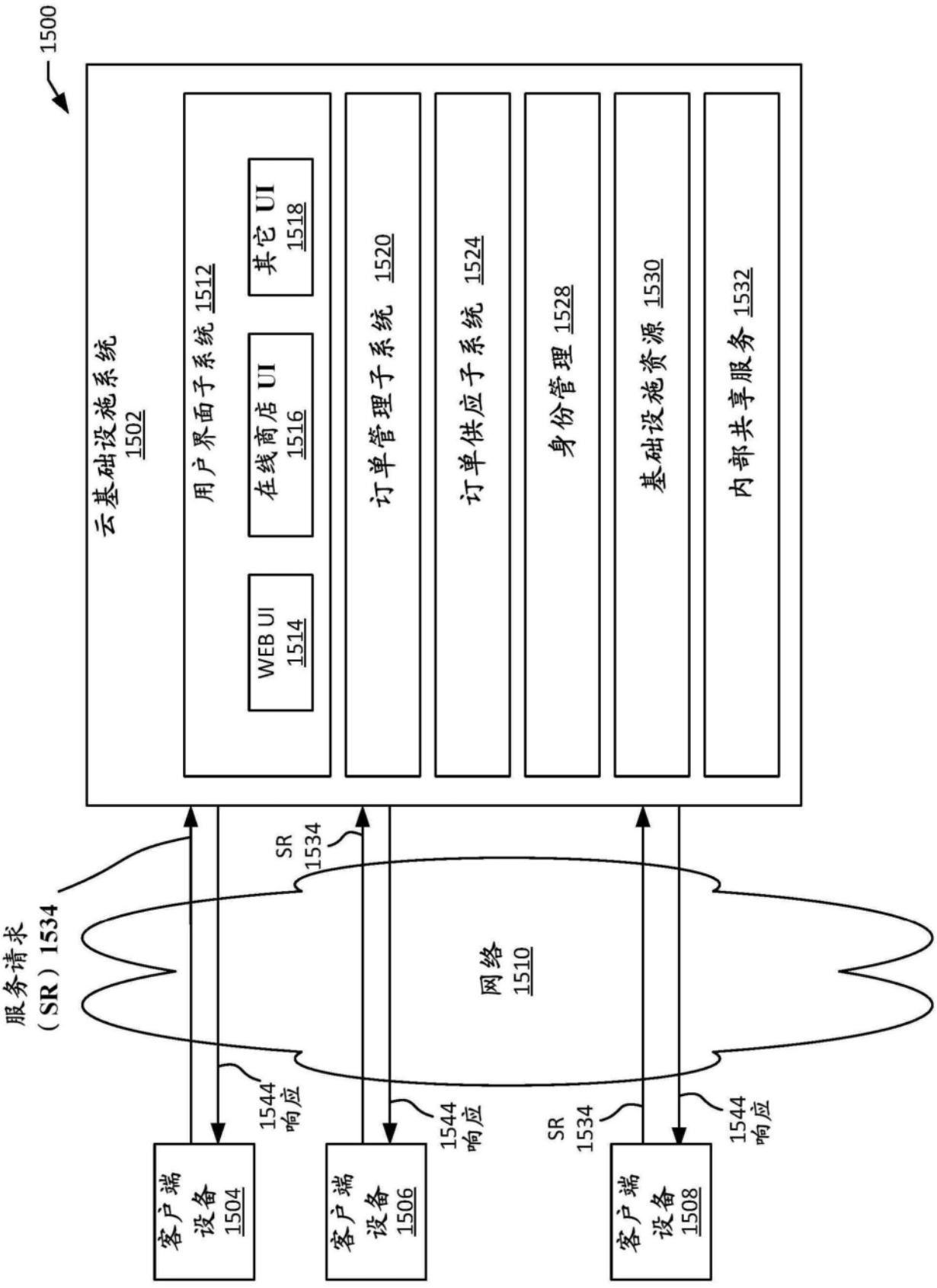

1500
云基础设施系统
1502
用户界面子系统 1512
WEB UI
1514
在线商店 UI
1516
其它 UI
1518
订单管理子系统 1520
订单供应子系统 1524
身份管理 1528
基础设施资源 1530
内部共享服务 1532
服务请求
（SR）1534
SR
1534
网络
1510
1544
响应
1544
响应
SR
1534
1544
响应
客户端
设备
1504
客户端
设备
1506
客户端
设备
1508

图15

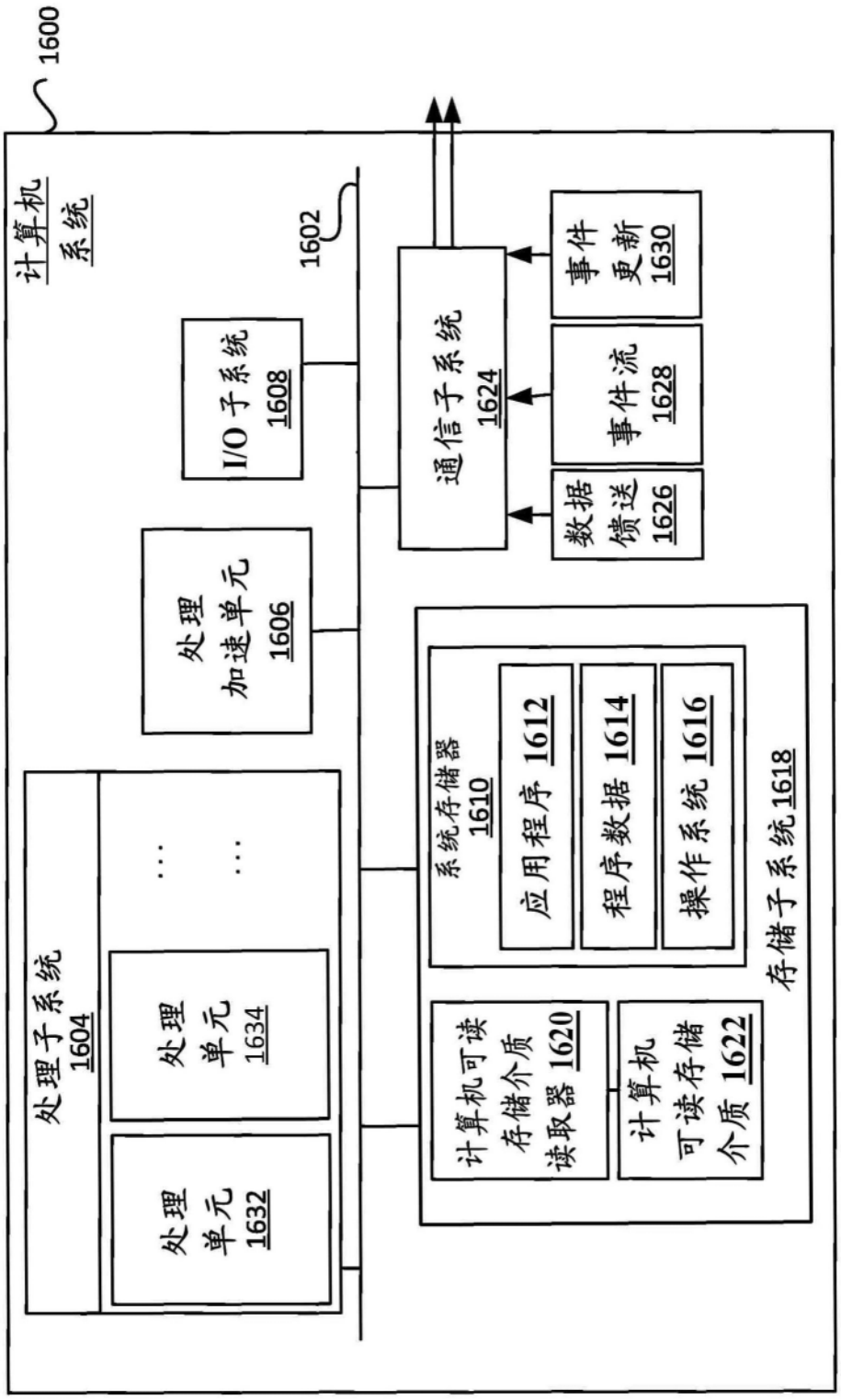

1600
计算机系统
1602
I/O 子系统 1608
处理加速单元 1606
通信子系统 1624
事件更新 1630
事件流 1628
数据馈送 1626
处理子系统 1604
处理单元 1632
处理单元 1634
...
...
系统存储器 1610
应用程序 1612
程序数据 1614
操作系统 1616
计算机可读存储介质读取器 1620
计算机可读存储介质 1622
存储子系统 1618

图16

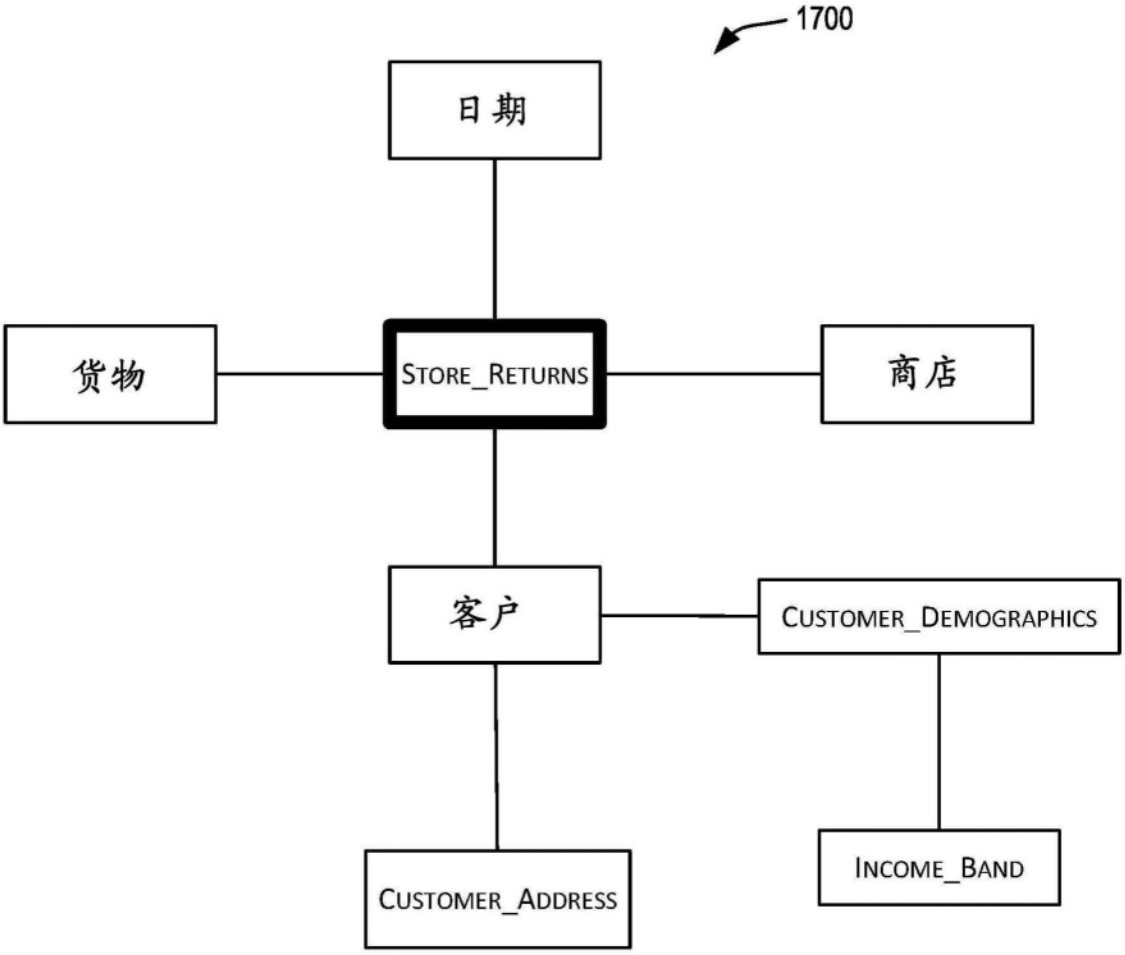

1700
日期
货物
STORE_RETURNS
商店
客户
CUSTOMER_DEMOGRAPHICS
CUSTOMER_ADDRESS
INCOME_BAND

图17