



(12) 发明专利

(10) 授权公告号 CN 108549666 B

(45) 授权公告日 2021.05.04

(21) 申请号 201810241432.X

(22) 申请日 2018.03.22

(65) 同一申请的已公布的文献号
申请公布号 CN 108549666 A

(43) 申请公布日 2018.09.18

(73) 专利权人 上海达梦数据库有限公司
地址 201203 上海市浦东新区博霞路50号
403室

(72) 发明人 张钦 宋鑫

(74) 专利代理机构 北京品源专利代理有限公司
11332

代理人 孟金喆

(51) Int. Cl.

G06F 16/2457 (2019.01)

G06F 16/2455 (2019.01)

(56) 对比文件

CN 107590161 A, 2018.01.16

CN 106201476 A, 2016.12.07

US 2014330801 A1, 2014.11.06

审查员 李兵兵

权利要求书3页 说明书17页 附图4页

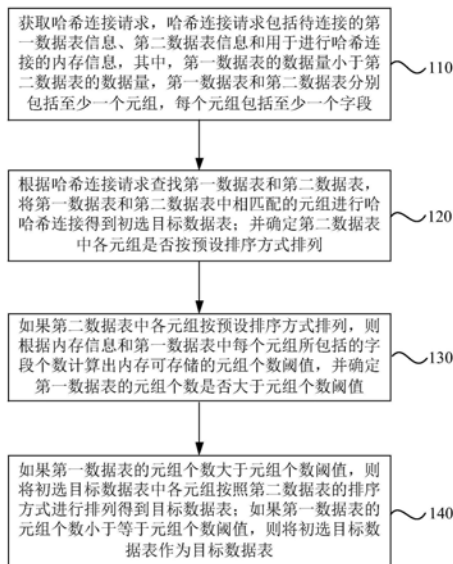
(54) 发明名称

一种数据表的排序方法、装置、设备及存储介质

(57) 摘要

本发明公开了一种数据表的排序方法、装置、设备及存储介质。该方法包括：获取哈希连接请求；根据哈希连接请求查找第一数据表和第二数据表，将第一数据表和第二数据表中相匹配的元组进行哈希连接得到初选目标数据表；并确定第二数据表中各元组是否按预设排序方式排列；如果第二数据表中各元组按预设排序方式排列，则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值，并确定第一数据表的元组个数是否大于元组个数阈值；如果第一数据表的元组个数大于元组个数阈值，则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表。本发明实施例减少了大量无用的排序操作，极大提高了数据库的运行效率。

CN 108549666 B



1. 一种数据表的排序方法,其特征在于,包括:

获取哈希连接请求,所述哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息,其中,所述第一数据表的数据量小于所述第二数据表的数据量,所述第一数据表和所述第二数据表分别包括至少一个元组,所述每个元组包括至少一个字段;

根据所述哈希连接请求查找所述第一数据表和所述第二数据表,将所述第一数据表和所述第二数据表中相匹配的元组进行哈希连接得到初选目标数据表;并确定所述第二数据表中各元组是否按预设排序方式排列;

如果所述第二数据表中各元组按预设排序方式排列,则根据所述内存信息和所述第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,并确定所述第一数据表的元组个数是否大于所述元组个数阈值;

如果所述第一数据表的元组个数大于所述元组个数阈值,则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到目标数据表;如果所述第一数据表的元组个数小于等于所述元组个数阈值,则将所述初选目标数据表作为目标数据表;

所述如果所述第二数据表中各元组按预设排序方式排列,则根据所述内存信息和所述第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,包括:

如果所述第二数据表中各元组按预设排序方式排列,则在预设的哈希连接操作符上设置自适应属性;如果所述第二数据表中各元组未按预设排序方式排列,则不在预设的哈希连接操作符上设置自适应属性;其中,所述自适应属性用于表示所述第二数据表中各元组按预设排序方式排列;

根据所述内存信息和所述第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值;

相应的,所述如果所述第一数据表的元组个数大于所述元组个数阈值,则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到目标数据表;如果所述第一数据表的元组个数小于等于所述元组个数阈值,则将所述初选目标数据表作为目标数据表,包括:

如果所述第一数据表的元组个数大于所述元组个数阈值,则取消所述预设的哈希连接操作符上的自适应属性;如果所述第一数据表的元组个数小于等于所述元组个数阈值,则保留所述预设的哈希连接操作符上的自适应属性;并确定所述预设的哈希连接操作符上是否保留所述自适应属性;

如果所述预设的哈希连接操作符上未保留所述自适应属性,则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到所述目标数据表;如果所述预设的哈希连接操作符上保留所述自适应属性,则将所述初选目标数据表作为所述目标数据表。

2. 根据权利要求1所述的方法,其特征在于,所述如果所述第二数据表中各元组按预设排序方式排列,则在预设的哈希连接操作符上设置自适应属性;如果所述第二数据表中各元组未按预设排序方式排列,则不在预设的哈希连接操作符上设置自适应属性;其中,所述自适应属性用于表示所述第二数据表中各元组按预设排序方式排列,包括:

如果所述第二数据表中各元组按预设排序方式排列,则分别在预设的哈希连接操作符和排序操作符上设置所述自适应属性;如果所述第二数据表中各元组未按预设排序方式排

列,则不在预设的哈希连接操作符和排序操作符上设置所述自适应属性;并确定所述预设的排序操作符上是否设置所述自适应属性;其中,所述自适应属性用于表示所述第二数据表中各元组按预设排序方式排列;

相应的,所述如果所述预设的哈希连接操作符上未保留所述自适应属性,则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到所述目标数据表;如果所述预设的哈希连接操作符上保留所述自适应属性,则将所述初选目标数据表作为所述目标数据表,包括:

如果所述预设的排序操作符上设置所述自适应属性,则确定所述预设的哈希连接操作符上是否保留所述自适应属性;

如果所述预设的哈希连接操作符上未保留所述自适应属性,则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到所述目标数据表;如果所述预设的哈希连接操作符上保留所述自适应属性,则将所述初选目标数据表作为所述目标数据表。

3. 根据权利要求2所述的方法,其特征在于,还包括:

如果所述预设的排序操作符上未设置所述自适应属性,则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到所述目标数据表。

4. 根据权利要求1-3任一项所述的方法,其特征在于,所述根据所述哈希连接请求查找所述第一数据表和所述第二数据表,将所述第一数据表和所述第二数据表中相匹配的元组进行哈希连接得到初选目标数据表,包括:

根据所述哈希连接请求查找所述第一数据表和所述第二数据表;

根据预设的哈希函数计算所述第一数据表中各元组的第一哈希键值,并根据所述第一哈希键值将所述第一数据表中各元组插入待填充表中对应的位置,得到哈希表;

按照预设顺序读取所述第二数据表中各元组,并根据所述预设的哈希函数计算所述第二数据表中各元组的第二哈希键值;

从所述哈希表中查找是否存在与所述第二哈希键值相匹配的所述第一哈希键值;

如果所述哈希表中存在与所述第二哈希键值相匹配的第一哈希键值,则将与所述第二哈希键值对应的所述第二数据表中的元组以及与所述第一哈希键值对应的所述哈希表中的元组进行哈希连接得到所述初选目标数据表。

5. 一种数据表的排序装置,其特征在于,包括:

哈希连接请求获取模块,用于获取哈希连接请求,所述哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息,其中,所述第一数据表的数据量小于所述第二数据表的数据量,所述第一数据表和所述第二数据表分别包括至少一个元组,所述每个元组包括至少一个字段;

初选目标数据表获取模块,用于根据所述哈希连接请求查找所述第一数据表和所述第二数据表,将所述第一数据表和所述第二数据表中相匹配的元组进行哈希连接得到初选目标数据表;并确定所述第二数据表中各元组是否按预设排序方式排列;

元组个数阈值确定模块,用于如果所述第二数据表中各元组按预设排序方式排列,则根据所述内存信息和所述第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,并确定所述第一数据表的元组个数是否大于所述元组个数阈值;

目标数据表获取模块,用于如果所述第一数据表的元组个数大于所述元组个数阈值,

则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到目标数据表；如果所述第一数据表的元组个数小于等于所述元组个数阈值，则将所述初选目标数据表作为目标数据表；

所述元组个数阈值确定模块，包括：

自适应属性设置子模块，用于如果所述第二数据表中各元组按预设排序方式排列，则在预设的哈希连接操作符上设置自适应属性；如果所述第二数据表中各元组未按预设排序方式排列，则不在预设的哈希连接操作符上设置自适应属性；其中，所述自适应属性用于表示所述第二数据表中各元组按预设排序方式排列；

元组个数阈值确定子模块，用于根据所述内存信息和所述第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值；

相应的，所述目标数据表获取模块，包括：

自适应属性检测子模块，用于如果所述第一数据表的元组个数大于所述元组个数阈值，则取消所述预设的哈希连接操作符上的自适应属性；如果所述第一数据表的元组个数小于等于所述元组个数阈值，则保留所述预设的哈希连接操作符上的自适应属性；并确定所述预设的哈希连接操作符上是否保留所述自适应属性；

目标数据表确定子模块，用于如果所述预设的哈希连接操作符上未保留所述自适应属性，则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到所述目标数据表；如果所述预设的哈希连接操作符上保留所述自适应属性，则将所述初选目标数据表作为所述目标数据表。

6. 根据权利要求5所述的装置，其特征在于，所述自适应属性设置子模块，包括：

自适应属性设置单元，用于如果所述第二数据表中各元组按预设排序方式排列，则分别在预设的哈希连接操作符和排序操作符上设置所述自适应属性；如果所述第二数据表中各元组未按预设排序方式排列，则不在预设的哈希连接操作符和排序操作符上设置所述自适应属性；并确定所述预设的排序操作符上是否设置所述自适应属性；其中，所述自适应属性用于表示所述第二数据表中各元组按预设排序方式排列；

相应的，所述目标数据表确定子模块，包括：

自适应属性检测单元，用于如果所述预设的排序操作符上设置所述自适应属性，则确定所述预设的哈希连接操作符上是否保留所述自适应属性；

目标数据表第一确定单元，用于如果所述预设的哈希连接操作符上未保留所述自适应属性，则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到所述目标数据表；如果所述预设的哈希连接操作符上保留所述自适应属性，则将所述初选目标数据表作为所述目标数据表。

7. 一种设备，其特征在于，包括：

一个或多个处理器；

存储器，用于存储一个或多个程序；

当所述一个或多个程序被所述一个或多个处理器执行，使得所述一个或多个处理器实现如权利要求1-4中任一所述的数据表的排序方法。

8. 一种计算机可读存储介质，其上存储有计算机程序，其特征在于，该程序被处理器执行时实现如权利要求1-4中任一所述的数据表的排序方法。

一种数据表的排序方法、装置、设备及存储介质

技术领域

[0001] 本发明实施例涉及数据库技术,尤其涉及一种数据表的排序方法、装置、设备及存储介质。

背景技术

[0002] 数据排序是数据库中的基本操作之一,用户可以通过如SQL (Structured Query Language, 结构化查询语言) 语句中的Order By语句指定需要对数据执行排序操作,当然,SQL语句中并没有要求必须执行数据排序操作,但是一些其它数据操作是要求在数据有序的基础上才可以执行的,因此,也可以理解为隐含有需要执行数据排序操作。因而,有必要对数据排序操作进行研究。由于数据表是由大量数据组成的,因此,对数据表进行排序操作归根到底是对数据表中的数据进行排序操作。

[0003] 数据排序操作通常由一个专门的操作符来执行,当数据量很大时,数据排序将是一个时间与空间消耗都很大的操作,而现有技术中,如果在数据处理过程中设定了需要执行数据排序操作,那么无论实际情况如何,都必须执行数据排序操作,但实际情况中可能存在下述情况:流向数据排序操作的数据,其本身已经按序排列,此时已经没有必要再执行数据排序操作,如,针对数据库中的两个或多个表进行表连接操作后得到的结果,其本身已经按序排列,这里的表连接可以是哈希连接。这种数据本身在进行数据排序操作之前,其本身就已经按序排列,这种情况下执行的数据排序操作可以理解为无用的操作,如果继续执行数据排序操作还会极大影响数据库的运行效率,因此,出于提高数据库运行效率的目的,有必要在满足数据库操作要求的基础上,尽可能的减少类似上述无用的数据排序操作。

发明内容

[0004] 本发明提供一种数据表的排序方法、装置、设备及存储介质,以实现提高数据库运行效率。

[0005] 第一方面,本发明实施例提供了一种数据表的排序方法,该方法包括:

[0006] 获取哈希连接请求,所述哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息,其中,所述第一数据表的数据量小于所述第二数据表的数据量,所述第一数据表和所述第二数据表分别包括至少一个元组,所述每个元组包括至少一个字段;

[0007] 根据所述哈希连接请求查找所述第一数据表和所述第二数据表,将所述第一数据表和所述第二数据表中相匹配的元组进行哈希连接得到初选目标数据表;并确定所述第二数据表中各元组是否按预设排序方式排列;

[0008] 如果所述第二数据表中各元组按预设排序方式排列,则根据所述内存信息和所述第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,并确定所述第一数据表的元组个数是否大于所述元组个数阈值;

[0009] 如果所述第一数据表的元组个数大于所述元组个数阈值,则将所述初选目标数据

表中各元组按照所述第二数据表的排序方式进行排列得到目标数据表;如果所述第一数据表的元组个数小于等于所述元组个数阈值,则将所述初选目标数据表作为目标数据表。

[0010] 第二方面,本发明实施例还提供了一种数据表的排序装置,该装置包括:

[0011] 哈希连接请求获取模块,用于获取哈希连接请求,所述哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息,其中,所述第一数据表的数据量小于所述第二数据表的数据量,所述第一数据表和所述第二数据表分别包括至少一个元组,所述每个元组包括至少一个字段;

[0012] 初选目标数据表获取模块,用于根据所述哈希连接请求查找所述第一数据表和所述第二数据表,将所述第一数据表和所述第二数据表中相匹配的元组进行哈希连接得到初选目标数据表;并确定所述第二数据表中各元组是否按预设排序方式排列;

[0013] 元组个数阈值确定模块,用于如果所述第二数据表中各元组按预设排序方式排列,则根据所述内存信息和所述第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,并确定所述第一数据表的元组个数是否大于所述元组个数阈值;

[0014] 目标数据表获取模块,用于如果所述第一数据表的元组个数大于所述元组个数阈值,则将所述初选目标数据表中各元组按照所述第二数据表的排序方式进行排列得到所述目标数据表;如果所述第一数据表的元组个数小于等于所述元组个数阈值,则将所述初选目标数据表作为目标数据表。

[0015] 第三方面,本发明实施例还提供了一种设备,该设备包括:

[0016] 一个或多个处理器;

[0017] 存储器,用于存储一个或多个程序;

[0018] 当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如前文所述的数据表的排序方法。

[0019] 第四方面,本发明实施例还提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如前文所述的数据表的排序方法。

[0020] 本发明通过获取哈希连接请求,根据哈希连接请求查找第一数据表和所述第二数据表,再将第一数据表和所述第二数据表中相匹配的元组进行哈希连接得到初选目标数据表,并确定第二数据表中各元组是否按预设排序方式排列,如果第二数据表中各元组按预设排序方式排列,则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,以及确定第一数据表的元组个数是否大于元组个数阈值,如果第一数据表的元组个数大于元组个数阈值,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表,如果第一数据表的元组个数小于等于元组个数阈值,则将初选目标数据表作为目标数据表,解决了现有技术中由于无法根据实际情况确定是否需要执行数据表的排序操作,导致产生了大量的无用操作,极大影响了数据库的运行效率的问题,实现了在满足数据库操作要求的基础上,可以根据实际情况确定是否执行数据表的排序操作,减少了大量的无用操作,极大地节省了时间与空间,提高了数据库的运行效率。

附图说明

[0021] 图1是本发明实施例一中的一种数据表的排序方法的流程图;

[0022] 图2是本发明实施例二中的一种数据表的排序方法的流程图;

[0023] 图3是本发明实施例三中的一种数据表的排序装置的结构示意图；

[0024] 图4是本发明实施例四中的一种设备的结构示意图。

具体实施方式

[0025] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是，此处所描述的具体实施例仅仅用于解释本发明，而非对本发明的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与本发明相关的部分而非全部结构。

[0026] 实施例一

[0027] 图1为本发明实施例一提供的一种数据表的排序方法的流程图，本实施例可适用于减少无用的排序操作的情况，该方法可以由数据表的排序装置来执行，该装置可以采用软件和/或硬件的方式实现，该装置可以配置于设备中，例如典型的是计算机等。如图1所示，该方法具体包括如下步骤：

[0028] 步骤110、获取哈希连接请求，哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息，其中，第一数据表的数据量小于第二数据表的数据量，第一数据表和第二数据表分别包括至少一个元组，每个元组包括至少一个字段。

[0029] 在本发明的具体实施例中，连接操作是基本关系数据库查询操作之一，是从两个关系的笛卡尔积中选取属性之间满足一定条件的元组，且是处理两个关系之间联系的唯一操作，其中，这里所述的关系可以理解为数据表；笛卡尔积是一个数据表的每一行依次与另一个数据表的所有行进行匹配。实现连接的方法较多，如嵌套循环连接、排列归并连接或哈希连接等，这些方法在不同的应用场合表现出不同的性能。其中，由于哈希连接可以解决嵌套循环连接中大量随机读取问题，以及排序合并连接中排序代价过大的问题，尤其适用于内存能够提供足够的空间，两个数据表的数据量较大的场合，此时，性能相对最优。为了便于后续理解，需要先对数据表的相关基本概念进行说明。笛卡尔积中每一个元素，如 (d_1, d_2, \dots, d_n) ，称为n元组或简称元组，以下统一称为元组，元组是关系数据库中的基本概念，更为具体的，数据表中的每行就是一个元组，或者，可以理解为数据表中的每行就是一条记录。数据表中的每列就是一个属性，或者，也可以理解为数据表中的每列就是一个字段。以下将数据表中的每行统一称为元组，每列统一称为字段。基于上述，可以理解到每个数据表可以包括至少一个元组；每个元组可以包括至少一个字段；笛卡尔积还可以是一个数据表的每个元组依次与另一个数据表中的元组进行匹配。

[0030] 下面对哈希连接进行说明，哈希连接是利用哈希表将两个数据表进行等值连接，也即上述所述的满足一定条件中的一些条件可以指等值连接，其原理为：构建哈希表，即在进行连接的两个数据表中，选取两个数据表中数据量较小的数据表，该数据表又称为内表，根据预设的哈希函数，计算该数据表中的元组的哈希键值，再根据哈希键值将相应的元组插入待填充表对应的位置，得到哈希表；查找哈希表，即选取两个数据表中数据量较大的数据表，该数据表又被称为外表，按照预设顺序，根据上述所述的哈希函数计算该数据表中每个元组的哈希键值，根据该哈希键值到构建好的哈希表中进行哈希查找，查找到满足等值连接的元组，将满足上述连接条件的元组进行哈希连接并输出，即如果数据量较大的数据表中的某个元组与哈希表中某个元组满足等值连接的条件，则将满足上述连接条件的元组进行哈希连接并输出。此外，还需要说明的是，在上述构建哈希表的过程中，根据计算得

到的哈希键值将数据量较小的数据表中的元组插入到待填充表中对应的位置得到哈希表后,需要将哈希表保存至内存。

[0031] 当需要对两个数据表采用哈希连接方法进行查询操作时,获取哈希连接请求,其中,哈希连接请求可以包括待连接的两个数据表信息和用于进行哈希连接的内存信息,待连接两个数据表信息可以分别为第一数据表信息和第二数据表信息。第一数据表信息和第二数据表信息所起到的作用可以是:能根据该信息查找到对应的数据表,如数据表名称或数据表存储路径等。用于进行哈希连接的内存信息可以作为确定是否可以将构建的哈希表保存至该内存区域的依据之一,该内存信息可以包括内存大小。需要说明的是,上述数据表信息和内存信息包括的内容可以根据具体情况进行设定,在此不作具体限定。此外,第一数据表的数据量小于第二数据表的数据量,同时,根据前文所述的关于数据表所具有的基本属性,可以理解到,第一数据表和第二数据表也分别包括至少一个元组,每个元组包括至少一个字段。

[0032] 步骤120、根据哈希连接请求查找第一数据表和第二数据表,将第一数据表和第二数据表中相匹配的元组进行哈希连接得到初选目标数据表;并确定第二数据表中各元组是否按预设排序方式排列。

[0033] 在本发明的具体实施例中,由于哈希连接请求中包括第一数据表信息和第二数据表信息,因此,根据哈希连接请求可以查找到与数据表信息对应的第一数据表和第二数据表,再对查找到的第一数据表和第二数据表进行进一步的数据处理得到初选目标数据表,具体处理过程可以为:根据前文所述的哈希连接的原理可知,利用数据量较小的数据表来构建哈希表,利用数据量较大的数据表作为探测表,用于从哈希表中查找是否存在相匹配的元组。于是,针对待连接的第一数据表和第二数据表而言,由于第一数据表的数据量小于第二数据表的数据量,因此,可以利用第一数据表来构建哈希表,可以利用第二数据表作为探测表,用于从哈希表中查找是否存在相匹配的元组。更为具体的,构建哈希表,即根据预设的哈希函数,计算第一数据表中的元组的哈希键值,再根据哈希键值将相应的元组插入待填充表对应的位置,得到哈希表;查找哈希表,即按照预设顺序,根据上述所述的哈希函数计算第二数据表中每个元组的哈希键值,根据该哈希键值到构建好的哈希表中进行哈希查找,查找到满足等值连接的元组,将满足上述连接条件的元组进行哈希连接并输出,即如果第二数据表中的某个元组与哈希表中某个元组满足等值连接的条件,则将满足上述连接条件的元组进行哈希连接并输出,将上述输出的连接结果作为初选目标数据保存至初选目标数据表中。此外,需要说明的是,可以根据设定的等值连接条件,只对第一数据表和第二数据表中的预设字段中相匹配的元组进行哈希连接,如,第一数据表有A、B、C和D四个字段,第二数据表有E、F和G三个字段,等值连接条件为A和F字段的元组,基于上述,这里的预设字段可以理解为第一数据表中的A字段和第二数据表中的F字段,相应的,可以只对第一数据表中的A字段和第二数据表中的F字段中相匹配的元组进行哈希连接。此外,还需要说明的是,上述如果针对的是第一数据表和第二数据表的所有字段,也可以将预设字段理解为所有字段。

[0034] 在上述过程中,还需要确定第二数据表中各元组是否按预设排序方式排列,这是由于如果第二数据表中各元组是预设排序方式排列的,那么得到的初选目标数据表中各元组也就有可能是按序排列的,这里所述的按序排列可以为升序,也可以为降序,具体按序排

列的设定可以根据实际情况进行确定,在此不作具体限定。其中,上述所说的初选目标数据表中各元组有可能是按序排列的,说明还存在即使第二数据表中各字段数值是按序排列的,得到的初选目标数据表中各元组未按序排列的情况。这种情况,是由哈希表超出了用于进行哈希连接的内存区域的内存大小导致的。综上,可以将第二数据表中各元组是否按预设排序方式排列作为得到的初选目标数据表中各元组是否按序排列所必须满足的条件之一,以便于再结合其它条件进一步确定是否需要按对初选目标数据表中各元组按照预设方式进行排序,并且如果对初选目标数据表中各元组是否按序排列所需满足的条件进行等级划分,那么第二数据表中各元组是否按预设排序方式排列的等级是最高的。相应的,可以理解到,如果第二数据表中各元组未按预设排序方式排列,那么可以直接确定初选目标数据表中各元组也未按序排列,需要进一步对初选目标数据表中各元组按照预设方式进行排序。

[0035] 此外,需要说明的是,上述所述预设排序方式可以是指定字段或指定组合字段按序排列,如每个元组包括A、B和C三个字段,当预设排序方式为指定字段按序排列时,这里的指定字段可以为字段A,相应的,各元组是否按预设排序方式排列可以指各元组是否按指定字段A的排序方式排列;或者这里的指定字段还可以为字段B,相应的,各元组是否按预设排序方式排列可以指各元组是否按指定字段B的排序方式排列;又或者这里的指定字段又可以为字段C,相应的,各元组是否按预设方式排列可以指各元组是否按指定字段C的排序方式排列。当预设排序方式为指定组合字段按序排列时,这里的指定组合字段可以为字段A和B,相应的,各元组是否按预设排序方式排列可以指各元组是否按指定组合字段A和B的排序方式排列;或者,这里的指定组合字段还可以为字段A和C,相应的,各元组是否按预设排序方式排列可以指各元组是否按指定组合字段A和C的排序方式排列;又或者,这里的指定组合字段又可以为字段B和C,相应的,各元组是否按预设排序方式排列可以指各元组是否按指定组合字段B和C的排序方式排列;又或者,这里的指定组合字段又可以为字段A、B和C,相应的,各元组是否按预设排序方式排列可以指各元组是否按指定组合字段A、B和C的排序方式排列。更为具体的,可以在Order By语句中设置各元组按指定字段或指定组合字段的排序方式进行排列,如“select form tbl1order by col1desc,col2,col3”,表示针对tbl1中各元组,先按照字段col1的数值降序排列,如果字段col1的数值相同,则按照字段col2的数值升序排列,如果字段col1和col2的数值均相同,则按照字段col3的数值升序排列。当然可以理解到,指定字段或指定组合字段按序排列中的按序排列也指可以为升序,还可以为降序。需要说明的是,上述预设排序方式可以根据实际情况进行设定,在此不作具体限定。

[0036] 此外,需要说明的是,这里所述的预设字段与指定字段并不相同,预设字段是在执行哈希连接操作过程中设定的,指定字段是在执行排序操作过程中设定的,其中,预设字段是根据等值连接条件确定出的第一数据表中的某个字段和第二数据表中某个字段,而指定字段是确定第二数据表中各元组是否按预设排序方式排列时设定的。示例性的,如前所述,假设第一数据表有A、B、C和D四个字段,第二数据表有E、F和G三个字段,执行哈希连接操作时,设定的等值连接条件为A和F字段的元组,因此,预设字段为第一数据表中的A字段和第二数据表中的F字段;执行排序操作时,设定预设排序方式为指定第二数据表中的字段F按序排列,因此,确定第二数据表中各元组是否按预设排序方式排列就需要确定第二数据表中各元组是否按指定字段F按序排列,具体的按序排列可以是升序,也可以是降序。

[0037] 通过根据哈希连接请求查找第一数据表和第二数据表,将第一数据表和第二数据

表中相匹配的元组进行哈希连接得到初选目标数据表,可以为后续得到目标数据表提供数据支持;通过确定第二数据表中各元组是否按预设排序方式排列,为后续确定是否需要初选目标数据表中各元组进行排序提供依据。

[0038] 步骤130、如果第二数据表中各元组按预设排序方式排列,则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,并确定第一数据表的元组个数是否大于元组个数阈值。

[0039] 在本发明的具体实施例中,根据前文所述可知,根据第一数据表构建得到的哈希表会存储到用于进行哈希连接的内存区域中,如果第一数据表的数据量大于内存大小,便会将该哈希表存储到磁盘上,再按照前述方式根据前述剩余的第一数据表重新构建新的哈希表,如果前述剩余的第一数据表的数据量也大于内存大小,也便会将该新的哈希表存储到磁盘上,再继续按照前述方式根据前述剩余的第一数据表重新构建新的哈希表,如果前述剩余的第一数据表的数据量又大于内存大小,继续会将该新的哈希表存储到磁盘上,直至前述剩余的第一数据表的数据量小于等于内存大小,即完成将第一数据表的各元组均插入构建得到的哈希表中。从上述可以理解到,第一数据表的各元组被存储到了多张哈希表中,相应的,第二数据表中的各元组被划分为与第一数据表对应的多个部分,即各个部分的元组需要从多张哈希表中进行哈希值匹配操作,在上述情况下,即使第二数据表中各元组是按预设排序方式排列的,也无法保证得到的初选目标数据表中各元组也是按序排列的,即存在着即使第二数据表中各元组是按预设排序方式排列的,得到的初选目标数据表中各元组并未按序排列,上述也即为步骤120中提及的存在的即使第二数据表中各元组是预设排序方式排列的,得到的初选目标数据表中各元组未按序排列的情况。

[0040] 根据上述可知,如何确定第一数据表的数据量是否大于内存大小便成为了关键,可以采用如下方式进行确定,具体的:数据量可以用元组个数和字段个数表示,哈希连接请求中包括的用于进行哈希连接的内存信息中包括内存大小,又由于在根据第一数据表构建哈希表的过程中,是依次计算各个元组的哈希键的,再根据哈希键值将相应的元组插入待填充表对应的位置,最终得到哈希表,也即第一数据表中的元组是一个一个插入待填充表的,相应的,哈希表也是逐渐形成的,与之对应的内存区域也是逐渐被占用的。基于此,可以将内存区域可存储的最多元组个数作为确定标准,更为具体的,可以根据内存信息和第一数据表中每个元组所包括的字段个数计算出该内存区域可存储的最多元组个数,并将最多元组个数作为元组个数阈值。之后,可以再比较第一数据表的元组个数与元组个数阈值的大小关系,如果确定出第一数据表的元组个数大于元组个数阈值,那么就可以说明第一数据表的各元组被存储到了多张哈希表中,更进一步的可以说明存在着即使第二数据表中各元组是按预设排序方式排列的,也无法保证得到的初选目标数据表中各元组也是按序排列的情况,在上述情况下,需要进一步对初选目标数据表中各元组按照预设方式进行排序。综上,可以将第一数据表的元组个数是否大于元组个数阈值作为得到的初选目标数据表中各元组是否按序排列所必须满足的条件之一,并且如果对初选目标数据表中各元组是否按序排列所需满足的条件进行等级划分,那么第一数据表的元组个数是否大于元组个数阈值的等级是低于步骤120所述的第二数据表中各元组是否按预设排序方式排列的等级的,也即为了确定是否需要初选目标数据表中各元组按照预设方式进行排序,首先,确定第二数据表中各元组是否按预设排序方式排列,如果确定出第二数据表中各元组按预设排序方式

排列,则接下来继续确定第一数据表的元组个数是否大于元组个数阈值,再基于第一数据表的元组个数与元组个数阈值的比较结果,确定是否需要初选目标数据表中各元组按照预设方式进行排序。

[0041] 需要说明的是,上述进行确定第一数据表的元组个数是否大于元组个数阈值的操作是基于第二数据表中各元组已是按预设排序方式排列的情况。

[0042] 步骤140、如果第一数据表的元组个数大于元组个数阈值,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果第一数据表的元组个数小于等于元组个数阈值,则将初选目标数据表作为目标数据表。

[0043] 在本发明的具体实施例中,如果确定出第一数据表的元组个数大于元组个数阈值,则可以说明第一数据表的各元组被存储到了多张哈希表中,相应的,第二数据表中的各元组被划分为与第一数据表对应的多个部分,即各个部分的元组需要从多张哈希表中进行哈希值匹配操作,在上述情况下,即使第二数据表中各元组是按预设排序方式排列的,也无法保证得到的初选目标数据表中各元组也是按序排列的,即存在着即使第二数据表中各元组是按预设排序方式排列的,得到的初选目标数据表中各元组并未按序排列,因此,需要进一步将初选目标数据表中各元组按照第二数据表的排序方式进行排列以得到目标数据表。相应的,如果确定出第一数据表的元组个数小于等于元组个数阈值,则可以说明第一数据表的各元组被存储到了同一张哈希表中,在此情况下,如果确定出第二数据表中各元组按预设排序方式排列,则可以确定得到的初选目标数据表中各元组也是按序排列的,因此,便无需再执行初选目标数据表中各元组按照第二数据表的排序方式进行排列的操作,直接将初选目标数据表作为目标数据表即可。

[0044] 基于上述,可以理解到,采用上述操作可以实现根据初选目标数据表的实际情况来确定是否需要执行对初选目标数据表中各元组按照第二数据表的排序方式进行排列的操作,即在已经可以确定初选目标数据表中各元组是按序排列的,便可以不再执行对初选目标数据表中各元组进行排序的操作,这种初选目标数据表在未执行排序操作之前,其本身就已经按序排列,这种情况下再执行的排序操作便可以理解为无用的操作,如果继续执行排序操作还会极大影响数据库的运行效率。因此,上述可以实现在满足数据库操作要求的基础上,减少上述无用的排序操作,进而大幅度节省时间与空间,提高数据库运行效率。而现有技术中,如果在数据表的处理过程中预先设定了执行排序操作,那么无论实际情况如何,都必须执行排序操作,也即现有技术中存在着执行无用操作的缺陷,上述操作也正是可以解决现有技术中存在的缺陷。

[0045] 此外,需要说明的是,对初选目标数据表中各元组还可以其它排序方式进行排列从而得到目标数据表,这里所述的其它排序方式可以为预设排列顺序。

[0046] 本实施例的技术方案,通过获取哈希连接请求,根据哈希连接请求查找第一数据表和第二数据表,再将第一数据表和第二数据表中相匹配的元组进行哈希连接得到初选目标数据表,并确定第二数据表中各元组是否按预设排序方式排列,如果第二数据表中各元组按预设排序方式排列,则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,以及确定第一数据表的元组个数是否大于元组个数阈值,如果第一数据表的元组个数大于元组个数阈值,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表,如果第一数据表的元组个数小于等于元组个数

阈值,则将初选目标数据表作为目标数据表,解决了现有技术中由于无法根据实际情况确定是否需要执行数据表的排序操作,导致产生了大量的无用操作,极大影响了数据库的运行效率的问题,实现了在满足数据库操作要求的基础上,可以根据实际情况确定是否执行数据表的排序操作,减少了大量的无用操作,极大地节省了时间与空间,提高了数据库的运行效率。

[0047] 可选的,在上述技术方案的基础上,如果第二数据表中各元组按预设排序方式排列,则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,具体可以包括:

[0048] 如果第二数据表中各元组按预设排序方式排列,则在预设的哈希连接操作符上设置自适应属性;如果第二数据表中各元组未按预设排序方式排列,则不在预设的哈希连接操作符上设置自适应属性;其中,自适应属性用于表示第二数据表中各元组按预设排序方式排列。

[0049] 根据内存信息和第一数据表中每个字段所包括的字段个数计算出内存可存储的元组个数阈值。

[0050] 在本发明的具体实施例中,由于关系数据库在执行SQL语句时,真正的物理执行计划都是通过操作符的实现来完成,因此,可以理解到,前文所述的对数据表执行的哈希连接以及排序操作便可以理解为设计哈希连接操作符和排序操作符的过程。相应的,关系数据库中数据表的连接可以指一个SQL语句通过数据表与数据表之间的关联,从一个或多个数据表检索出相关的元组。连接是通过SQL语句中From从句的多个数据表名称,以及Where从句里定义的数据表之间的连接条件来实现的。

[0051] 具体的,如果确定出第二数据表中各元组按预设排序方式排列,则可以在预设的哈希连接操作符上自适应属性;相反的,如果确定出第二数据表中各元组未按预设排序方式排列,则可以不在预设的哈希连接操作符上设置自适应属性,其中,自适应属性可以用于表示第二数据表中各元组是按预设排序方式排列的。这里所述的自适应属性也可以理解为标识,即可以根据确定预设的哈希连接操作符上是否保留该标识来确定第二数据表中各元组是否按预设排序方式排列。上述为后续确定是否需要初选目标数据表中各元组执行排序操作提供了依据。

[0052] 相应的,如果第一数据表的元组个数大于元组个数阈值,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果第一数据表的元组个数小于等于元组个数阈值,则将初选目标数据表作为目标数据表,具体可以包括:

[0053] 如果第一数据表的元组个数大于元组个数阈值,则取消预设的哈希连接操作符上的自适应属性;如果第一数据表的元组个数小于等于元组个数阈值,则保留预设的哈希连接操作符上的自适应属性;并确定预设的哈希连接操作符上是否保留自适应属性。

[0054] 如果预设的哈希连接操作符上未保留自适应属性,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果预设的哈希连接操作符上保留自适应属性,则将初选目标数据表作为目标数据表。

[0055] 在本发明的具体实施例中,如果第一数据表的元组个数大于元组个数阈值,则可以说明第一数据表的各元组被存储到了多张哈希表中,相应的,第二数据表中的各元组被划分为与第一数据表对应的多个部分,即各个部分的元组需要从多张哈希表中进行哈希值

匹配操作,在上述情况下,即使第二数据表中各元组是按预设排序方式排列的,也无法保证得到的初选目标数据表中各元组也是按序排列的,即存在着即使第二数据表中各元组是按预设排序方式排列的,得到的初选目标数据表中各元组并未按序排列,因此,需要进一步将初选目标数据表中各元组按照第二数据表的排序方式进行排列以得到目标数据表。基于上述,便可以取消预设的哈希连接操作符上的自适应属性。相反的,如果第一数据表的元组个数小于等于元组个数阈值,则可以说明第一数据表的各元组被存储到了同一张哈希表中,在此情况下,如果确定出第二数据表中各元组按预设排序方式排列,则可以确定得到的初选目标数据表中各元组也是按序排列的,因此,便无需再执行初选目标数据表中各元组按照第二数据表的排序方式进行排列的操作,直接将初选目标数据表作为目标数据表即可。基于上述,便可以继续保留预设的哈希连接操作符上设置的自适应属性。

[0056] 进而,在上述基础上,如果确定出预设的哈希连接操作符上未保留自适应属性,则说明需要将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表。相反的,如果确定出预设的哈希连接操作符上保留自适应属性,则说明无需再对初选目标数据表进行排序,直接将初选目标数据表作为目标数据表即可。

[0057] 可选的,在上述技术方案的基础上,如果第二数据表中各元组按预设排序方式排列,则在预设的哈希连接操作符上设置自适应属性;如果第二数据表中各元组未按预设排序方式排列,则不在预设的哈希连接操作符上设置自适应属性;其中,所述自适应属性用于表示所述第二数据表中各元组按预设排序方式排列,具体可以包括:

[0058] 如果第二数据表中各元组按预设排序方式排列,则分别在预设的哈希连接操作符上和排序操作符上设置自适应属性;如果第二数据表中各元组未按预设排序方式排列,则不在预设的哈希连接操作符和排列操作符上设置自适应属性;并确定预设的排序操作符上是否设置自适应属性;其中,所述自适应属性用于表示所述第二数据表中各元组按预设排序方式排列。

[0059] 在本发明的具体实施例中,如果确定出第二数据表中各元组按预设排序方式排列,则可以分别在预设的哈希连接操作符和排序操作符上自适应属性;相反的,如果确定出第二数据表中各元组未按预设排序方式排列,则可以不在预设的哈希连接操作符和排序操作符上设置自适应属性,即可以根据确定预设的哈希连接操作符是否保留自适应属性和预设的排序操作符上是否设置该标识来确定第二数据表中各元组是否按预设排序方式排列。上述为后续确定是否需要初选目标数据表中各元组执行排序操作提供了依据。当然,需要说明的是,确定预设的哈希连接操作符是否保留自适应属性和预设的排序操作符上是否设置自适应属性的执行是有先后顺序的,其中,先执行确定预设的排序操作符上是否设置自适应属性,再根据确定出的结果,确定是否执行确定预设的哈希连接操作符上是否保留自适应属性。这是由于预设的排序操作上是否设置自适应属性只与第二数据表中各元组是否按预设排序方式排列有关,如果确定出第二数据表中各元组未按预设排序方式排列,那么就可以确定初选目标数据表中各元组也是未按序排列的,此时,需要进一步对初选目标数据表执行排序操作;而虽然同样预设的哈希连接操作符上是否设置自适应属性只与第二数据表中各元组是否按预设排序方式排列有关,但是进一步的,预设的哈希连接操作符上是否保留自适应属性与第一数据表的元组个数是否大于元组个数阈值有关,而确定第一数据表的元组个数是否大于元组个数阈值的前提是已经确定出第二数据表中各元组按预设

排序方式排列,也即确定预设的哈希连接操作符上是否保留自适应属性的前提是预设的哈希连接操作符上已经设置了自适应属性。这也可以理解为确定是否保留自适应属性的前提是已经设置了自适应属性。

[0060] 相应的,如果预设的哈希连接操作符上未保留自适应属性,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果预设的哈希连接操作符上保留自适应属性,则将初选目标数据表作为目标数据表,具体可以包括:

[0061] 如果预设的排序操作符上设置自适应属性,则确定预设的哈希连接操作符上是否保留自适应属性。

[0062] 如果预设的哈希连接操作符上未保留自适应属性,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果预设的哈希连接操作符上保留自适应属性,则将初选目标数据表作为目标数据表。

[0063] 在本发明的具体实施例中,如果确定出预设的排序操作符上设置自适应属性,则可以说明已经确定出第二数据表中各元组按预设排序方式排列,基于此,继续确定预设的哈希连接操作符上是否保留自适应属性,如前文所述,如果确定出预设的哈希连接操作符上未保留自适应属性,则说明需要将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表。相反的,如果确定出预设的哈希连接操作符上保留自适应属性,则说明无需再对初选目标数据表进行排序,直接将初选目标数据表作为目标数据表即可。

[0064] 可选的,在上述技术方案的基础上,该方法具体还可以包括:

[0065] 如果预设的排序操作符上未设置自适应属性,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表。

[0066] 在本发明的具体实施例中,如果确定出预设的排序操作符上未设置自适应属性,则可以说明第二数据表中各元组未按预设排序方式排列,也就同时说明初选目标数据表中各元组未按预设排序方式排列,基于此,需要对初选目标数据表执行排序操作,即将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表。

[0067] 可选的,在上述技术方案的基础上,根据哈希连接请求查找第一数据表和第二数据表,将第一数据表和第二数据表中相匹配的元组进行哈希连接得到初选目标数据表,具体可以包括:

[0068] 根据哈希连接请求查找第一数据表和第二数据表。

[0069] 根据预设的哈希函数计算第一数据表中各元组的第一哈希键值,并根据第一哈希键值将第一数据表中各元组插入待填充表中对应的位置,得到哈希表。

[0070] 按照预设顺序读取第二数据表中各元组,并根据预设的哈希函数计算第二数据表中各元组的第二哈希键值。

[0071] 从哈希表中查找是否存在于第二哈希键值相匹配的第一哈希键值。

[0072] 如果哈希表中存在与第二哈希键值相匹配的第一哈希键值,则将第二哈希键值对应的第二数据表中的元组以及与第一哈希键值对应的哈希表中的元组进行哈希连接得到初选目标数据表。

[0073] 在本发明的具体实施例中,哈希函数是计算机科学中经常使用到的一种处理函数,利用哈希键值的快速搜索算法已经被认为是成熟的检索手段。哈希函数处理过的数据特征是“相同数值的哈希函数值一定相同,不同数值的哈希函数值可能相同”,具体的,预设

的哈希函数可以为have_fun_1和have_fun_2,当然可以理解到,哈希函数可以根据实际情况进行设定,在此不作具体限定。根据预设的哈希函数计算第一数据表中的各元组的第一哈希键值,再根据第一哈希键值将相应的元组插入待填充表对应的位置,得到哈希表,之后按照预设顺序读取第二数据表中各元组,并根据上述所述的哈希函数计算该数据表中每个元组的第二哈希键值,根据第二哈希键值到构建好的哈希表中进行哈希查找,查找到满足等值连接的元组,将满足上述连接条件的元组进行哈希连接并输出,得到初选目标数据表,即如果第二数据表中的某个元组与哈希表中某个元组满足等值连接的条件,则将满足上述连接条件的元组进行哈希连接并输出,得到初选目标数据表。需要说明的是,上述计算第一哈希键值和第二哈希键值所采用的哈希函数相同。

[0074] 实施例二

[0075] 图2为本发明实施例二提供的一种数据表的排序方法的流程图,本实施例可适用于减少无用的排序操作的情况,该方法可以由数据表的排序装置来执行,该装置可以采用软件和/或硬件的方式实现,该装置可以配置于设备中,例如典型的是计算机等。如图2所示,该方法具体包括如下步骤:

[0076] 步骤201、获取哈希连接请求,哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息,其中,第一数据表的数据量小于第二数据表的数据量,第一数据表和第二数据表分别包括至少一个元组,每个元组包括至少一个字段。

[0077] 步骤202、根据哈希连接请求查找第一数据表和第二数据表。

[0078] 步骤203、根据预设的哈希函数计算第一数据表中各元组的第一哈希键值,并根据第一哈希键值将第一数据表中各元组插入待填充表中对应的位置,得到哈希表。

[0079] 步骤204、按照预设顺序读取第二数据表中的一个元组作为当前处理元组,并根据预设的哈希函数计算第二数据表中当前处理元组的第二哈希键值。

[0080] 步骤205、从哈希表中查找是否存在与第二哈希键值相匹配的第一哈希键值;若是,则执行步骤206;若否,则转入执行步骤207。

[0081] 步骤206、将与第二哈希键值对应的第二数据表中的元组以及与第一哈希键值对应的哈希表中的元组进行哈希连接得到初选目标数据。

[0082] 步骤207、确定是否完成第二数据表中各元组的哈希键值匹配操作;若是,则执行步骤208;若否,则返回执行步骤204。

[0083] 步骤208、根据初选目标数据得到初选目标数据表。

[0084] 步骤209、确定第二数据表中各元组是否按预设排序方式排列;若是,则执行步骤210;若否,则执行步骤211。

[0085] 步骤210、在预设的哈希连接操作符和排序操作符上设置自适应属性;转入执行步骤212。

[0086] 步骤211、不在预设的哈希键值操作符和排序操作符上设置自适应属性;转入执行步骤217。

[0087] 步骤212、根据内存信息和第一数据表中每个元组所包括的字段个数计算内存可存储的元组个数阈值。

[0088] 步骤213、确定第一数据表的元组个数是否大于元组个数阈值;若是,则执行步骤215;若否,则执行步骤214。

[0089] 步骤214、保留预设的哈希连接操作符上的自适应属性；转入执行步骤216。

[0090] 步骤215、取消预设的哈希连接操作符上的自适应属性；转入执行步骤217。

[0091] 步骤216、将初选目标数据表作为目标数据表。

[0092] 步骤217、将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表。

[0093] 在本发明的具体实施例中，在步骤204中，按照预设顺序依次读取第二数据表中的每个元组作为当前处理元组，并根据预设的哈希函数计算该当前处理元组的第二哈希键值，转入执行步骤205，从哈希表中查找是否存在与该第二哈希键值相匹配的第一哈希键值，如果存在与该第二哈希键值相匹配的第一哈希键值，则执行步骤206，将与该第二哈希键值对应的第二数据表中的元组，也即当前处理元组，以及与第一哈希键值对应的哈希表中的元组就那些哈希连接得到一组初选目标数据，如果不存在与该第二哈希键值相匹配的第一哈希键值，则执行步骤207，确定是否已经完成了对第二数据表中所有元组执行的哈希键值匹配操作，即步骤204-步骤206，如果还没有完成对第二数据表中所有元组执行的哈希键值匹配操作，则返回执行步骤204，相应的，会循环执行步骤204-207，直至确定完成了对第二数据表中各元组执行的哈希键值匹配操作，便可以跳出该循环，再依次执行下面的步骤。

[0094] 此外，在确定是否需要初选目标数据表执行排序操作时，在确定预设的排序操作符上设置有自适应属性的基础上，可以根据确定预设的哈希连接操作符上是否保留自适应属性来进一步确定是否需要初选目标数据表执行排序操作，具体的，如果确定出预设的哈希连接操作符上保留自适应属性，则将初选目标数据表作为目标数据表；如果确定出预设的哈希连接操作符上未保留自适应属性，则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表。

[0095] 本实施例的技术方案，通过采用上述操作可以实现根据初选目标数据表的实际情况来确定是否需要执行对初选目标数据表中各元组按照第二数据表的排序方式进行排列的操作，即在已经可以确定初选目标数据表中各元组是按序排列的，便可以不再执行对初选目标数据表中各元组进行排序的操作，这种初选目标数据表在未执行排序操作之前，其本身就已经按序排列，这种情况下再执行的排序操作便可以理解为无用的操作，如果继续执行排序操作还会极大影响数据库的运行效率。因此，上述可以在满足数据库操作要求的基础上，减少上述无用的排序操作，进而大幅度节省时间与空间，提高数据库运行效率。而现有技术中，如果在数据表的处理过程中预先设定了执行排序操作，那么无论实际情况如何，都必须执行排序操作，也即现有技术中存在着执行无用操作的缺陷，上述操作也正是可以解决现有技术中存在的缺陷。

[0096] 实施例三

[0097] 图3为本发明实施例三提供的一种数据表的检测装置的结构示意图，本实施例可适用于减少无用的排序操作的情况，该装置可以采用软件和/或硬件的方式实现，该装置可以配置于设备中，例如典型的是计算机等。如图3所示，该装置具体包括：

[0098] 哈希连接请求获取模块310，用于获取哈希连接请求，哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息，其中，第一数据表的数据量小于第二数据表的数据量，第一数据表和第三数据表分别包括至少一个元组，每个元

组包括至少一个字段；

[0099] 初选目标数据表获取模块320,用于根据哈希连接请求查找第一数据表和第二数据表,将第一数据表和第二数据表中相匹配的元组进行哈希连接得到初选目标数据表;并确定第二数据表中各元组是否按预设排序方式排列;

[0100] 元组个数阈值确定模块330,用于如果第二数据表中各元组按预设排序方式排列,则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,并确定第一数据表的元组个数是否大于元组个数阈值;

[0101] 目标数据表获取模块340,用于如果第一数据表的元组个数大于元组个数阈值,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果第一数据表的元组个数小于等于元组个数阈值,则将初选目标数据表作为目标数据表。

[0102] 本实施例的技术方案,通过哈希连接请求获取模块310获取哈希连接请求,初选目标数据表获取模块320根据哈希连接请求查找第一数据表和第二数据表,再将第一数据表和第二数据表中相匹配的元组进行哈希连接得到初选目标数据表,并确定第二数据表中各元组是否按预设排序方式排列,元组个数阈值确定模块330如果第二数据表中各元组按预设排序方式排列,则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,以及确定第一数据表的元组个数是否大于元组个数阈值,目标数据表获取模块340如果第一数据表的元组个数大于元组个数阈值,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表,如果第一数据表的元组个数小于等于元组个数阈值,则将初选目标数据表作为目标数据表,解决了现有技术中由于无法根据实际情况确定是否需要执行数据表的排序操作,导致产生了大量的无用操作,极大影响了数据库的运行效率的问题,实现了在满足数据库操作要求的基础上,可以根据实际情况确定是否执行数据表的排序操作,减少了大量的无用操作,极大地节省了时间与空间,提高了数据库的运行效率。

[0103] 可选的,在上述技术方案的基础上,元组个数阈值确定模块330,具体可以包括:

[0104] 自适应属性设置子模块,用于如果第二数据表中各元组按预设排序方式排列,则在预设的哈希连接操作符上设置自适应属性;如果第二数据表中各元组未按预设排序方式排列,则不在预设的哈希连接操作符上设置自适应属性;其中,自适应属性用于表示第二数据表中各元组按预设排序方式排列;

[0105] 元组个数阈值确定子模块,用于根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值;

[0106] 相应的,目标数据表获取模块340,具体可以包括:

[0107] 自适应属性检测子模块,用于如果第一数据表的元组个数大于元组个数阈值,则取消预设的哈希连接操作符上的自适应属性;如果第一数据表的元组个数小于等于元组个数阈值,则保留预设的哈希连接操作符上的自适应属性;并确定预设的哈希连接操作符上是否保留自适应属性;

[0108] 目标数据表确定子模块,用于如果预设的哈希连接操作符上未保留自适应属性,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果预设的哈希连接操作符上保留自适应属性,则将初选目标数据表作为目标数据表。

[0109] 可选的,在上述技术方案的基础上,自适应属性设置子模块,具体可以包括:

[0110] 自适应属性设置单元,用于如果第二数据表中各元组按预设排序方式排列,则分别在预设的哈希连接操作符和排序操作符上设置自适应属性;如果第二数据表中各元组未按预设排序方式排列,则不在预设的哈希连接操作符和排序操作符上设置自适应属性;并确定预设的排序操作符上是否设置自适应属性;其中,自适应属性用于表示第二数据表中各元组按预设排序方式排列;

[0111] 相应的,目标数据表确定子模块,具体可以包括:

[0112] 自适应属性检测单元,用于如果预设的排序操作符上设置自适应属性,则确定预设的哈希连接操作符上是否保留自适应属性;

[0113] 目标数据表第一确定单元,用于如果预设的哈希连接操作符上未保留自适应属性,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果预设的哈希连接操作符上保留自适应属性,则将初选目标数据表作为目标数据表。

[0114] 可选的,在上述技术方案的基础上,该方法具体还可以包括:

[0115] 目标数据表第二确定单元,用于如果预设的排序操作符上未设置自适应属性,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表。

[0116] 可选的,在上述技术方案的基础上,初选目标数据表获取模块320,具体可以包括:

[0117] 数据表查找子模块,用于根据哈希连接请求查找第一数据表和第二数据表;

[0118] 哈希表获取子模块,用于根据预设的哈希函数计算第一数据表中各元组的第一哈希键值,并根据第一哈希键值将第一数据表中各元组插入待填充表中对应的位置,得到哈希表;

[0119] 第二哈希键值计算子模块,用于按照预设顺序读取第二数据表中各元组,并根据预设的哈希函数计算第二数据表中各元组的第二哈希键值;

[0120] 哈希键值匹配子模块,用于从哈希表中查找是否存在与第二哈希键值相匹配的第一哈希键值;

[0121] 初选目标数据表第二获取子模块,用于如果哈希表中存在与第二哈希键值相匹配的第一哈希键值,则将与第二哈希键值对应的第二数据表中的元组以及与第一哈希键值对应的第一数据表中的元组进行哈希连接得到初选目标数据表。

[0122] 本发明实施例所提供的配置于设备的数据表的排序装置可执行本发明任意实施例所提供的应用于设备的数据表的排序方法,具备执行方法相应的功能模块和有益效果。

[0123] 实施例四

[0124] 图4为本发明实施例四提供的一种设备的结构示意图。图4示出了适于用来实现本发明实施方式的示例性设备412的框图。图4显示的设备412仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0125] 如图4所示,设备412以通用计算设备的形式表现。设备412的组件可以包括但不限于:一个或者多个处理器416,系统存储器428,连接于不同系统组件(包括系统存储器428和处理器416)的总线418。

[0126] 总线418表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器,外围总线,图形加速端口,处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说,这些体系结构包括但不限于工业标准体系结构 (ISA) 总线,微通道体系结构 (MAC) 总线,增强型ISA总线、视频电子标准协会 (VESA) 局域总线以及外围组件互连 (PCI) 总线。

[0127] 设备412典型地包括多种计算机系统可读介质。这些介质可以是任何能够被设备412访问的可用介质,包括易失性和非易失性介质,可移动的和不可移动的介质。

[0128] 系统存储器428可以包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器(RAM)430和/或高速缓存存储器432。设备412可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例,存储系统434可以用于读写不可移动的、非易失性磁介质(图4未显示,通常称为“硬盘驱动器”)。尽管图4中未示出,可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器,以及对可移动非易失性光盘(例如CD-ROM,DVD-ROM或者其它光介质)读写的光盘驱动器。在这些情况下,每个驱动器可以通过一个或者多个数据介质接口与总线418相连。存储器428可以包括至少一个程序产品,该程序产品具有一组(例如至少一个)程序模块,这些程序模块被配置以执行本发明各实施例的功能。

[0129] 具有一组(至少一个)程序模块442的程序/实用工具440,可以存储在例如存储器428中,这样的程序模块442包括但不限于操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块442通常执行本发明所描述的实施例中的功能和/或方法。

[0130] 设备412也可以与一个或多个外部设备414(例如键盘、指向设备、显示器424等)通信,还可与一个或者多个使得用户能与该设备412交互的设备通信,和/或与使得该设备412能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口422进行。并且,设备412还可以通过网络适配器420与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,如因特网)通信。如图所示,网络适配器420通过总线418与设备412的其它模块通信。应当明白,尽管图4中未示出,可以结合设备412使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0131] 处理器416通过运行存储在系统存储器428中的程序,从而执行各种功能应用以及数据处理,例如实现本发明实施例所提供的一种支付方式配置方法,包括:

[0132] 获取哈希连接请求,哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息,其中,第一数据的数据量小于第二数据表的数据量,第一数据表和第二数据表分别包括至少一个元组,每个元组包括至少一个字段。

[0133] 根据哈希连接请求查找第一数据表和第二数据表,将第一数据表和第二数据表中相匹配的元组进行哈希连接得到初选目标数据表;并确定第二数据表中各元组是否按预设排序方式排列。

[0134] 如果第二数据表中各元组按预设排序方式排列,则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,并确定第一数据表的元组个数是否大于元组个数阈值。

[0135] 如果第一数据表的元组个数大于元组个数阈值,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果第一数据表的元组个数小于等于元组个数阈值,则将初选目标数据表作为目标数据表。

[0136] 实施例五

[0137] 本发明实施例五还提供了一种计算机可读存储介质,其上存储有计算机程序,该

程序被处理器执行时实现如本发明实施例所提供的一种数据表的排序方法,该方法包括:

[0138] 获取哈希连接请求,哈希连接请求包括待连接的第一数据表信息、第二数据表信息和用于进行哈希连接的内存信息,其中,第一数据的数据量小于第二数据表的数据量,第一数据表和第二数据表分别包括至少一个元组,每个元组包括至少一个字段。

[0139] 根据哈希连接请求查找第一数据表和第二数据表,将第一数据表和第二数据表中相匹配的元组进行哈希连接得到初选目标数据表;并确定第二数据表中各元组是否按预设排序方式排列。

[0140] 如果第二数据表中各元组按预设排序方式排列,则根据内存信息和第一数据表中每个元组所包括的字段个数计算出内存可存储的元组个数阈值,并确定第一数据表的元组个数是否大于元组个数阈值。

[0141] 如果第一数据表的元组个数大于元组个数阈值,则将初选目标数据表中各元组按照第二数据表的排序方式进行排列得到目标数据表;如果第一数据表的元组个数小于等于元组个数阈值,则将初选目标数据表作为目标数据表。

[0142] 本发明实施例的计算机存储介质,可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是一—但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式计算机磁盘、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)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0143] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0144] 计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括——但不限于无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0145] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码,所述程序设计语言包括面向对象的程序设计语言——诸如Java、Smalltalk、C++、Ruby、Go,还包括常规的过程式程序设计语言——诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、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。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)——连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0146] 注意,上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解,本发明不限于这里所述的特定实施例,对本领域技术人员来说能够进行各种明显的变化、

重新调整和替代而不会脱离本发明的保护范围。因此,虽然通过以上实施例对本发明进行了较为详细的说明,但是本发明不仅仅限于以上实施例,在不脱离本发明构思的情况下,还可以包括更多其他等效实施例,而本发明的范围由所附的权利要求范围决定。

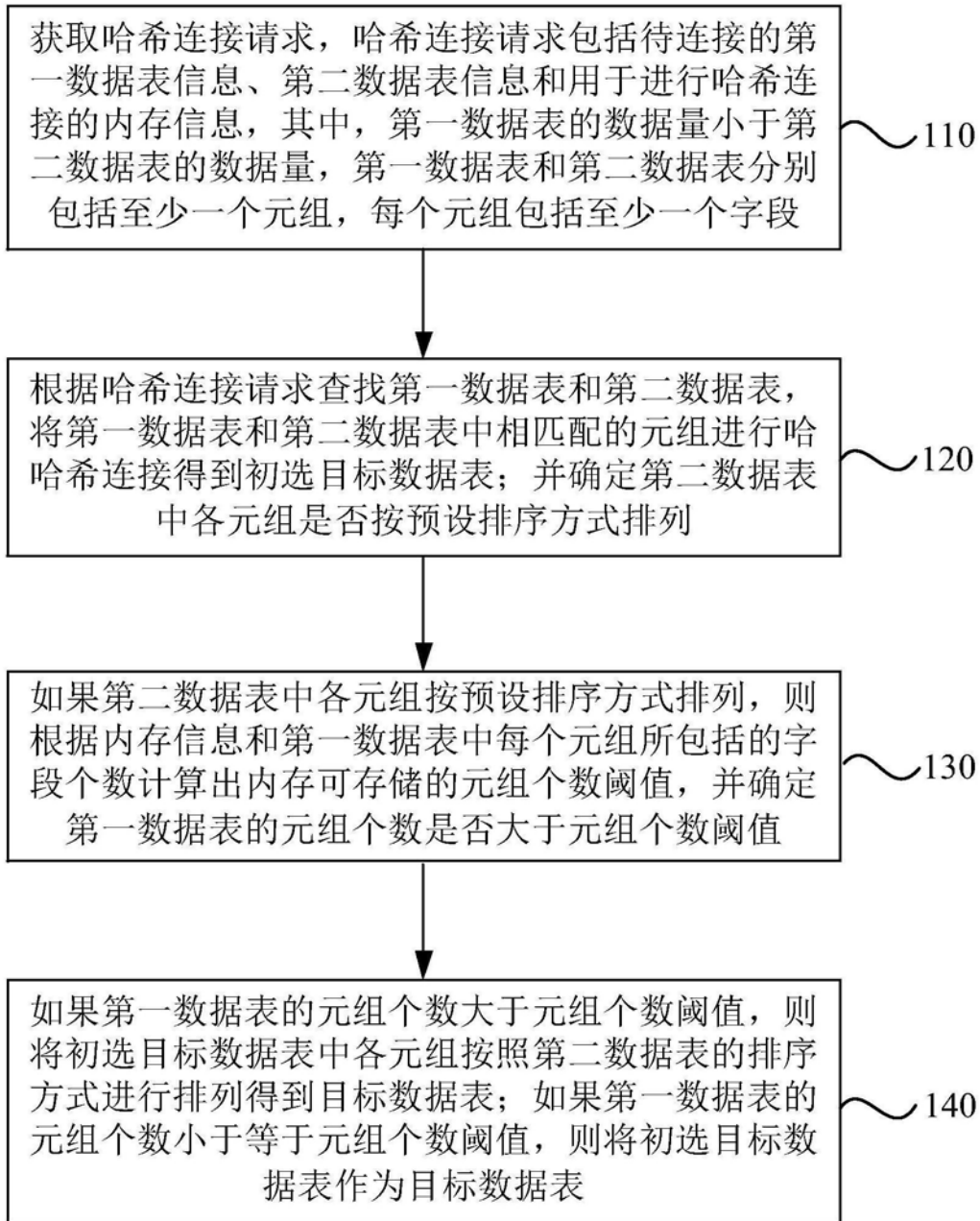


图1

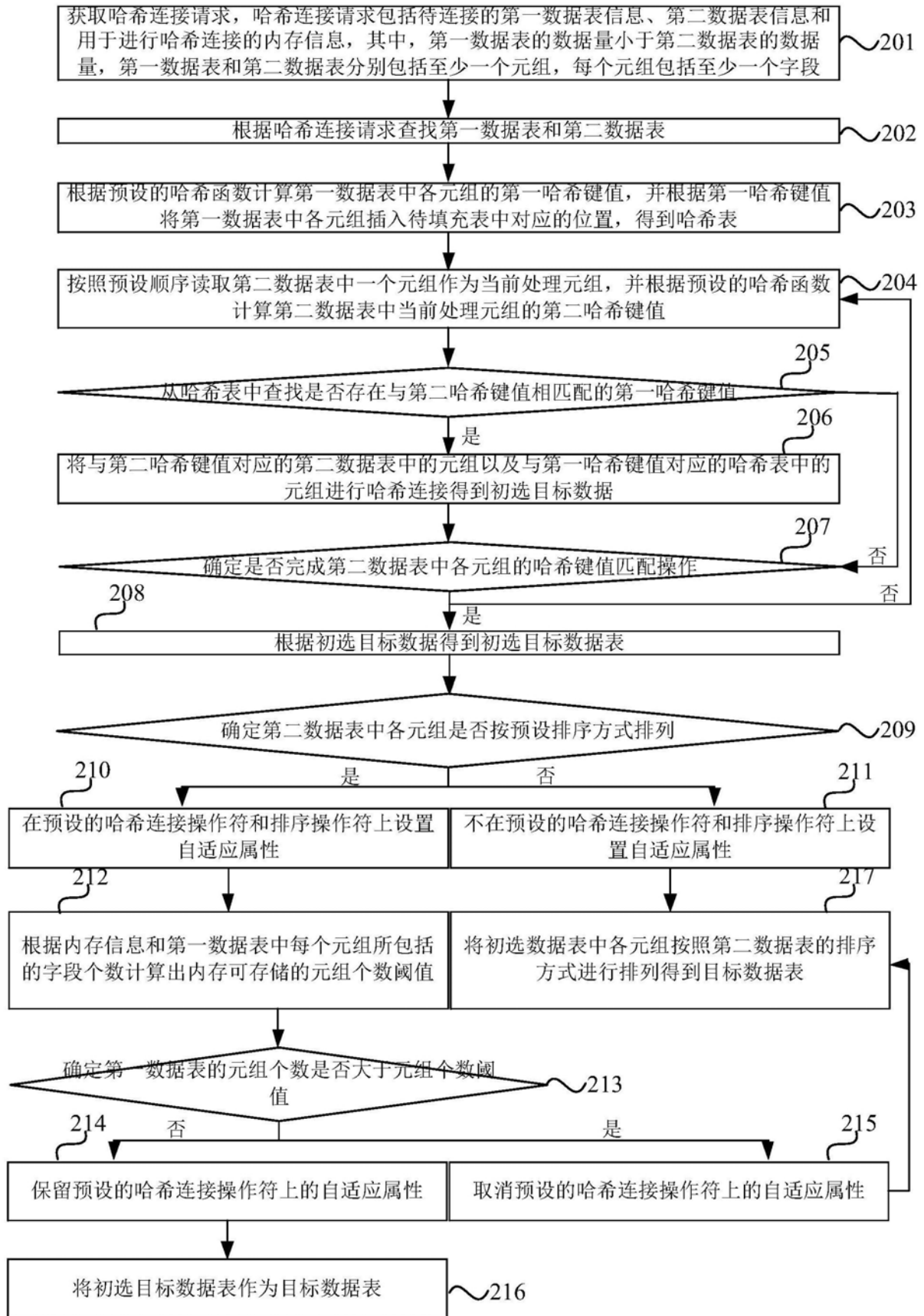


图2

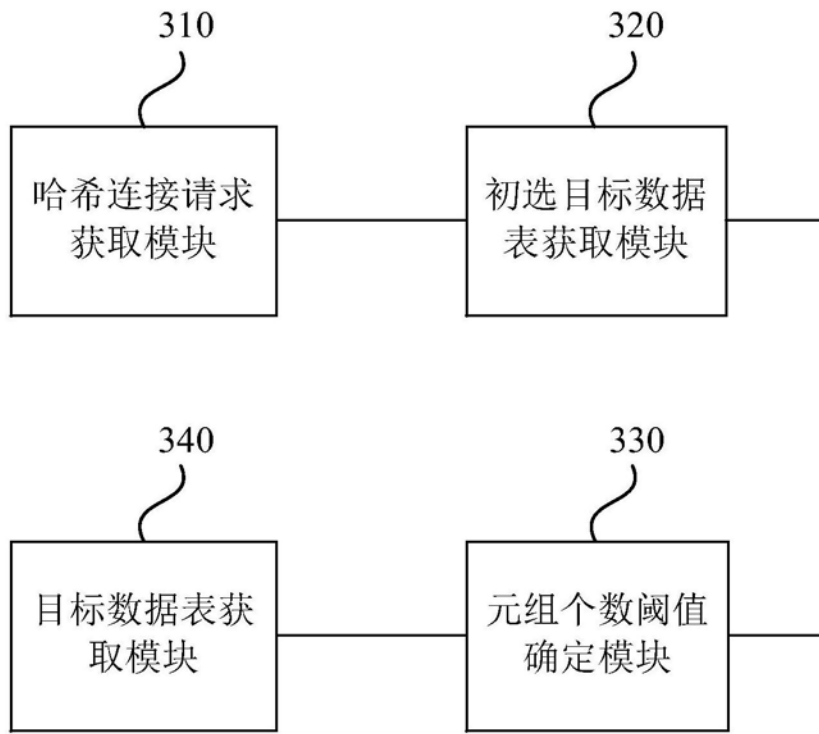


图3

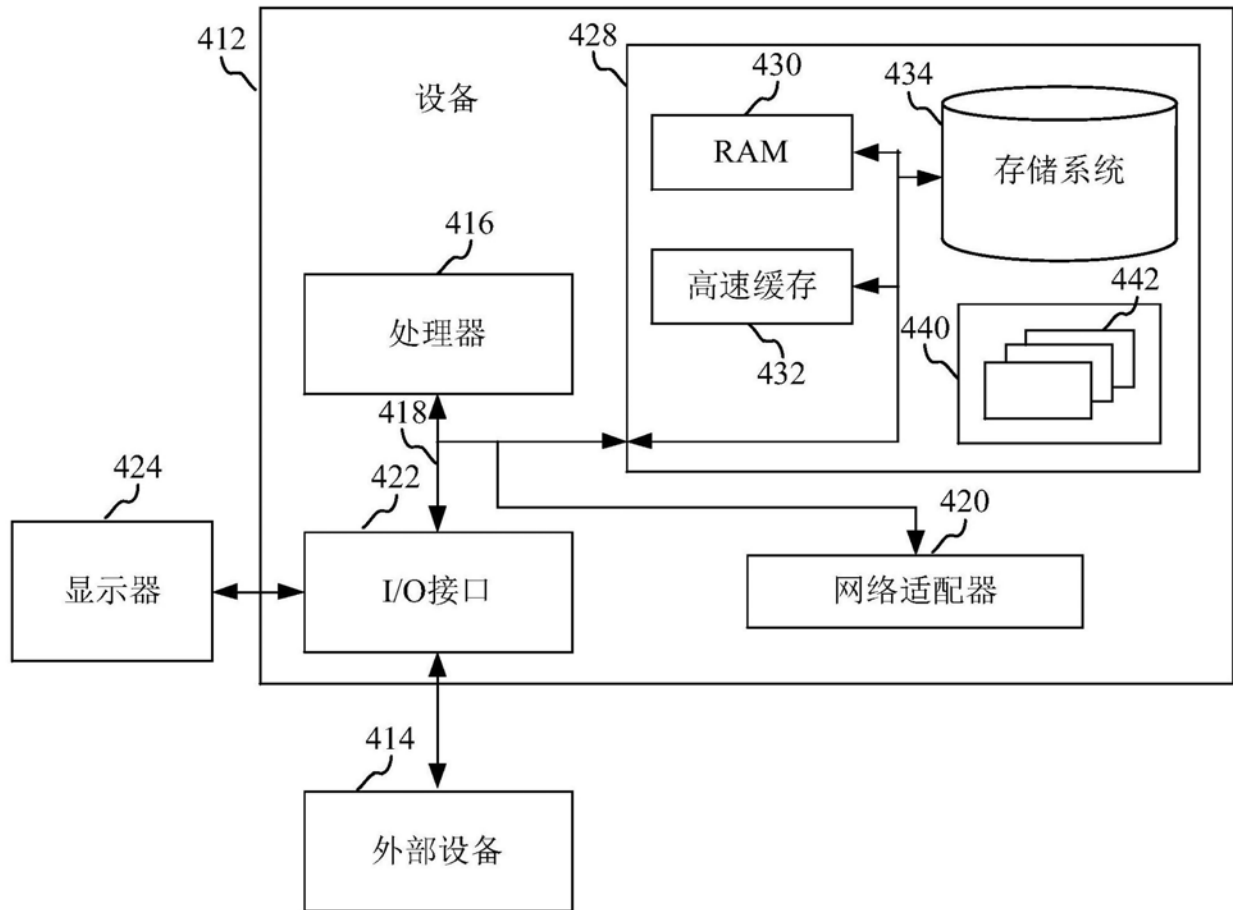


图4