

## 数据库表的数据重整方法和装置、介质和计算机设备

### 技术领域

[01]本公开涉及数据库技术领域，尤其涉及数据库表的数据重整方法和装置、介质和计算机设备。

### 背景技术

[02]在数据库中，常常需要对数据库表的存储格式进行更改。在更改存储格式时，需要对数据库表进行数据重整，即，按照新的存储格式将数据库表中的数据重写一遍。相关技术中一般使用普通事务插入的方式进行数据重整，这种方式开销较大。

### 发明内容

[03]第一方面，本公开实施例提供一种数据库表的数据重整方法，所述方法包括：响应于对源表的 DDL 操作，读取所述源表中的数据；所述 DDL 操作能够改变所述源表的存储结构；对读取的数据进行排序，将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，并记录各个数据块在目标表所在的数据库节点的磁盘的存储地址；在各个数据块均写入完成之后，基于各个数据块的存储地址生成所述目标表的 SSTable，所述目标表的 SSTable 包括所述各个数据块。

[04]在一些实施例中，所述源表和所述目标表分别为不同的命名空间中的数据库表，且所述源表所在的命名空间与所述目标表所在的命名空间相互隔离；所述方法还包括：在基于各个数据块的存储地址生成所述目标表的 SSTable 之后，将所述目标表移动到所述源表所在的命名空间。

[05]在一些实施例中，每个命名空间包括一个标志位，标志位不同的命名空间相互隔离；所述将所述目标表移动到所述源表所在的命名空间，包括：将所述目标表的名称更改为所述源表的名称；以及将所述目标表所在的命名空间的标志位更改为与所述源表所在的命名空间的标志位相同的标志位。

[06]在一些实施例中，所述目标表所在的命名空间中的表为不可见状态。

[07]在一些实施例中，所述方法还包括：在将所述目标表移动到所述源表所在的命名空间之后，删除所述源表。

[08]在一些实施例中，所述目标表包括多个目标表副本，所述多个目标表副本位于分布式数据库中的多个数据库节点上，所述多个数据库节点包括主节点和至少一个备份节点；

所述将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，包括：将排序后的数据按数据块写入所述主节点的目标表副本，并将写入所述主节点的目标表副本的数据块写入所述主节点的重做日志；将所述主节点的重做日志同步到各个备份节点的重做日志，以使各个备份节点基于本节点的重做日志将排序后的数据按数据块写入本节点的磁盘。

[09]在一些实施例中，所述源表包括多个第一分区，不同的第一分区存储在分布式数据库中不同的第一数据库节点上；所述对读取的数据进行排序，包括：分别读取各个第一数据库节点中存储的第一分区的数据；对从各个第一数据库节点读取的第一分区的数据进行排序。

[10]在一些实施例中，所述目标表包括多个第二分区，不同的第二分区存储在分布式数据库中不同的第二数据库节点上；所述将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，包括：基于排序后的数据所属的第二分区，将排序后的数据按数据块写入目标表在的第二数据库节点的磁盘。

[11]在一些实施例中，所述源表包括第一目标项，所述第一目标项包括所述源表的约束信息和/或外键；所述方法还包括：基于所述目标表的 SSTable 批量读取所述目标表中对应于所述第一目标项的第二目标项；每次读取的第二目标项对应于所述目标表中至少一种类型的数据；基于读取的第二目标项对应的数据生成校验向量；一个校验向量中包括的各个元素为同一类型的数据；基于每种类型的数据对应的约束信息对读取的第二目标项中对应类型的数据的校验向量进行校验。

[12]在一些实施例中，所述源表包括触发器；所述将所述目标表的名称更改为所述源表的名称，包括：为所述源表和所述目标表设置写锁，并在所述源表和所述目标表均设置成功之后将所述目标表的名称更改为所述源表的名称；所述方法还包括：在所述目标表中创建触发器；其中，在所述目标表中创建触发器的操作、为所述源表设置写锁的操作、将所述目标表的名称更改为所述源表的名称的操作，以及将所述目标表所在的命名空间的标志位更改为与所述源表所在的命名空间的标志位相同的标志位的操作包括在相同的事务中。

[13]在一些实施例中，所述目标表包括多个目标表副本，所述多个目标表副本位于分布式数据库中的多个数据库节点上，所述多个数据库节点包括主节点和至少一个备份节点；所述方法还包括：创建所述主节点的目标表副本的索引；将所述目标表副本的索引写入所述主节点的重做日志；将所述主节点的重做日志同步到各个备用节点，以使所述各个

备用节点基于本节点的重做日志创建本节点的目标表副本的索引。

[14]在一些实施例中，各个数据块在目标表所在的数据库节点的磁盘的存储地址记录在所述目标表所在的数据库节点的内存中的数据存储结构中。

[15]第二方面，本公开实施例提供一种数据库表的数据重整装置，所述装置包括：读取模块，用于响应于对源表的 DDL 操作，读取所述源表中的数据；所述 DDL 操作能够改变所述源表的存储结构；记录模块，用于对读取的数据进行排序，将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，并记录各个数据块在目标表所在的数据库节点的磁盘的存储地址；生成模块，用于在各个数据块均写入完成之后，基于各个数据块的存储地址生成所述目标表的 SSTable，所述目标表的 SSTable 包括所述各个数据块。

[16]第三方面，本公开实施例提供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任一实施例所述的方法。

[17]第四方面，本公开实施例提供一种计算机设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现任一实施例所述的方法。

[18]本公开实施例在源表的存储结构改变时，采用排序的方式对源表中的数据进行排序，并将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，由于每个数据块在磁盘中的存储地址都被记录下来了，因此，在各个数据块都写入完成之后，能够直接基于各个数据块的存储地址生成目标表的 SSTable，无需采用普通事务方式将数据写入内存，再将内存中的数据写入磁盘，从而节省了数据重整过程的开销，提高了数据重整效率。

[19]应当理解的是，以上的一般描述和后文的细节描述仅是示例性和解释性的，而非限制本公开。

#### 附图说明

[20]此处的附图被并入说明书中并构成本说明书的一部分，这些附图示出了符合本公开的实施例，并与说明书一起用于说明本公开的技术方案。

[21]图 1 是本公开实施例的数据库表的数据重整方法的流程图。

[22]图 2 是本公开实施例的数据库节点的数据存储结构的示意图。

[23]图 3 是本公开另一实施例的数据库节点的数据存储结构的示意图。

[24]图 4 是本公开实施例的分布式数据库的示意图。

[25]图 5 是本公开实施例的数据库表的数据重整装置的框图。

[26]图 6 是本公开实施例的计算机设备的结构示意图。

### 具体实施方式

[27]这里将详细地对示例性实施例进行说明，其示例表示在附图中。下面的描述涉及附图时，除非另有表示，不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本公开相一致的所有实施方式。相反，它们仅是与如所附权利要求书中所详述的、本公开的一些方面相一致的装置和方法的例子。

[28]在本公开使用的术语是仅仅出于描述特定实施例的目的，而非旨在限制本公开。在本公开和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式，除非上下文清楚地表示其他含义。还应当理解，本文中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。另外，本文中术语“至少一种”表示多种中的任意一种或多种中的至少两种的任意组合。

[29]应当理解，尽管在本公开可能采用术语第一、第二、第三等来描述各种信息，但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如，在不脱离本公开范围的情况下，第一信息也可以被称为第二信息，类似地，第二信息也可以被称为第一信息。取决于语境，如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

[30]为了使本技术领域的人员更好的理解本公开实施例中的技术方案，并使本公开实施例的上述目的、特征和优点能够更加明显易懂，下面结合附图对本公开实施例中的技术方案作进一步详细的说明。

[31]在数据库中，常常需要对数据库表的存储格式进行更改。需要更改存储格式的场景包括但不限于接收到对数据库表的修改主键、修改列类型、修改字符集等数据定义语言（Data Definition Language, DDL）操作。在更改存储格式时，需要对数据库表进行数据重整，即，按照新的存储格式将数据库表中的数据重写一遍。

[32]相关技术中一般使用普通事务插入的方式进行数据重整，这种普通事务方式需要先后执行将源表中的数据记录到内存、获取事务的版本号、并发控制、将内存中的数据写入磁盘等过程，该过程涉及内存中的数据与磁盘中的数据的多次合并（compaction），且采用普通事务的方式会按行对依赖对象进行处理，处理耗时较长。综上所述，采用普

通事务方式进行数据重整的开销较大。

[33]基于此，本公开实施例提供一种数据库表的数据重整方法，参见图 1，所述方法包括步骤 102 至步骤 106。

[34]步骤 102：响应于对源表的 DDL 操作，读取所述源表中的数据；所述 DDL 操作能够改变所述源表的存储结构；

[35]步骤 104：对读取的数据进行排序，将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，并记录各个数据块在目标表所在的数据库节点的磁盘的存储地址；

[36]步骤 106：在各个数据块均写入完成之后，基于各个数据块的存储地址生成所述目标表的 SSTable，所述目标表的 SSTable 包括所述各个数据块。

[37]本公开实施例在源表的存储结构改变时，采用排序的方式对源表中的数据进行排序，并将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，由于每个数据块在磁盘中的存储地址都被记录下来了，因此，在各个数据块都写入完成之后，能够直接基于各个数据块的存储地址生成目标表的 SSTable，无需采用普通事务方式将数据写入内存，再将内存中的数据写入磁盘，从而节省了数据重整过程的开销，提高了数据重整效率。下面对本公开实施例的方案进行具体说明。

[38]在步骤 102 中，源表可以是数据库中的任意的数据库表，例如，订单表，该订单表可以用于记录用户通过网络平台进行购物的时间、订单号、订单金额等信息。在一些实施例中，数据库为分布式数据库，则源表可以是分布式数据库的任意数据库节点上的数据库表。作为一种实现方式，源表可以包括多个分区（称为第一分区），不同的第一分区可以存储在分布式数据库的不同数据库节点上。源表包括的第一分区可以基于创建源表的 DDL 操作确定。假设源表包括 M 行 N 列，则可以划分为 k 个第一分区，每个第一分区包括源表中的 M/k 行 N/k 列。除了均匀地划分第一分区之外，不同的第一分区大小也可以不相等，本公开对此不做限制。

[39]可以针对源表执行 DDL 操作，该 DDL 操作可以是任意能够改变源表的存储结构的操作，例如，修改源表的主键、修改源表的列类型、修改源表的字符集等。可以响应于对源表的 DDL 操作读取源表中的数据。在源表存储在单个数据库节点上时，可以从源表所在的数据库节点读取源表中的数据。在源表包括多个第一分区，且不同的第一分区存储在不同的数据库节点上时，可以分别读取各个数据库节点上的第一分区，并将各个第一分区的数据整合起来，从而读取到源表中的数据。

[40]在步骤 104 中，可以基于对源表的 DDL 操作对从源表中读取的数据进行排序。在所述源表包括多个第一分区，且不同的第一分区存储在分布式数据库中不同的第一数据库节点上的情况下，可以分别读取各个第一数据库节点中存储的第一分区的数据，并对从各个第一数据库节点读取的第一分区的数据进行排序。该 DDL 操作可以改变源表的存储结构，例如，改变源表中至少两行和/或至少两列的存储顺序。以源表是订单表为例，假设该订单表中的各个订单是按照订单金额从大到小的顺序进行排序的，则一种 DDL 操作可以是将订单表中的各个订单按照订单金额从小到大的顺序进行排序。或者，假设该订单表的主键为订单号，且订单表中的各个字段从左到右依次为订单号、订单金额和下单时间，则一种 DDL 操作可以是将订单表中的下单时间作为主键，则排序后的各个字段从左到右依次为下单时间、订单号和订单金额。或者，一种 DDL 操作可以是更改源表包括的第一分区的数量。可以理解，上述仅为示例性说明，本公开实施例的排序方式不限于此。

[41]在对读取的数据进行排序之后，可以将排序后的数据划分为一个或多个数据块，每个数据块可以包括至少一行排序后的数据。其中，每个数据块的大小可以是固定的，例如，16K 或 2M。在一些实施例中，可以将排序后的数据划分为一个或多个宏块，每个宏块的大小可以是 2M；进一步地，还可以把每个宏块划分为多个微块，每个微块的大小可以是 16K。在这种情况下，可以以宏块为单位，将排序后的数据写入目标表所在的数据库节点的磁盘。也就是说，每次都向目标表所在的数据库节点的磁盘写入一个 2M 大小的宏块。在一些实施例中，还可以将各个微块进行压缩后组织成宏块，每个宏块中包括的微块的数量可以是不固定的。除了上述划分数据块的方式之外，还可以采用其他方式将排序后的数据划分为数据块，例如，可以将排序后的数据划分为大小不等的多个数据块，对此本公开实施例不再一一举例说明。

[42]每写入一个数据块，可以记录该数据块在目标表所在的数据库节点的磁盘的存储地址。其中，目标表可以是预先建立的数据库表，也可以是响应于接收到对源表的 DDL 操作而建立的数据库表。为了简洁，下文将目标表所在的数据库节点的磁盘称为目标磁盘。各个数据块可以写入目标磁盘中连续或不连续的存储地址。其中，记录的存储地址，可以是该数据块在目标磁盘中的存储地址的偏移量，也可以是数据块在目标磁盘中的存储地址的实际存储地址。在数据块的大小不固定的情况下，还可以记录每个数据块的大小，并将数据块的大小与该数据块在目标磁盘的存储地址进行关联。

[43]在一些实施例中，所述目标表包括多个第二分区，不同的第二分区存储在分布式数

数据库中不同的第二数据库节点上。目标表包括的第二分区的数量可以基于对源表的 DDL 操作确定。在目标表包括多个第二分区的情况下，可以基于排序后的数据所属的第二分区，将排序后的数据按数据块写入目标表所在的第二数据库节点的磁盘。例如，假设排序后的数据中的某一列所属的第二分区位于数据库节点 A 上，则可以将该列数据写入数据库节点 A 的磁盘。

[44]在步骤 106 中，在写入最后一个数据块之后，可以生成包括各个数据块的 SSTable。在一些实施例中，数据库节点的数据存储结构可以基于 LSM Tree 实现，具体来说，参见图 2，一个数据库节点的数据存储结构自上而下依次包括内存表（memtable）、Minor SSTable 和 Major SSTable 三层数据结构。其中，内存表是 LSM Tree 中的内存数据结构，存储在数据库节点的内存中。Minor SSTable 是 LSM Tree 中持久化在磁盘中的数据结构，可以由多个宏块组成，数据量较小。Minor SSTable 可能有多个。Major SSTable 是 LSM Tree 中持久化在磁盘中的数据结构，位于 LSM Tree 的最底层，数据量较大。Major SSTable 一般只有一个。在包括上述数据存储结构的基础上，本步骤中生成的 SSTable 可以是目标表的 Major SSTable。由于本公开实施例中直接生成 LSM Tree 最底层的 Major SSTable，因此，减少了 LSM Tree 的合并开销，从而减少了 CPU 和 IO 的开销。

[45]数据库节点中存储的数据可以按照该数据库节点的数据存储结构自上而下的顺序被查询。例如，在从一个数据库节点的目标表中查询目标数据时，可以先从该数据库节点的内存表中查询目标数据。如果从内存表中未查询到目标数据，再从该数据库节点的 Minor SSTable 中查询目标数据。如果从该数据库节点的内存表和 minor SSTable 中均未查询到目标数据，再从该数据库节点的 major SSTable 中查询目标数据。

[46]在一些实施例中，用于生成目标表的 Major SSTable 的数据块可以通过压缩处理得到，例如，在前述实施例中，数据块可以是宏块，宏块中可以包括经压缩处理的多个微块。在这种情况下，一方面，通过数据压缩能够减少 Major SSTable 对磁盘空间的占用；另一方面，由于采用了三层数据存储结构，数据查询过程是按照数据库节点的数据存储结构自上而下的顺序进行的，因此，在查询过程中，会先在未被压缩的内存表和 minor SSTable 中进行查询，只有在内存表和 minor SSTable 中均未查询到相关数据时，才需要从 major SSTable 中查询数据，因此，即便 major SSTable 中的数据经过了压缩处理，对数据查询过程的影响也比较小。

[47]进一步地，为了便于记录存储地址，可以在上述三层数据存储结构之外再新增一层数据存储结构，用于记录各个数据块的存储地址，如图 3 所示。新增的数据存储结构可

以是内存中的数据存储结构，为了方便与内存中原有的 memtable 区分开来，新增的数据存储结构可以称为 DDL memtable。这样，在内存中仅需要记录各个数据块的存储地址，而不需要记录各个数据块本身，从而减少内存占用。

[48]在上述实施例中，从源表中读取的数据可以是源表中的存量数据，即，在对源表执行 DDL 操作之前被存储到源表中的数据。这些存量数据最终可以被写入目标表的 major SSTable。进一步地，源表中除了存量数据以外，还可以包括增量数据，即，在对源表执行 DDL 操作之后被存储到源表中的数据。其中，源表中的增量数据可以写入目标表所在的数据库节点的内存表和 minor SSTable，而无需写入各个数据库节点的 major SSTable。具体来说，可以将源表中的增量数据可以写入目标表所在的数据库节点的内存表，并将内存表转储为 minor SSTable。这样，在 major SSTable 中包括的数据块经过压缩处理的情况下，无需每次写入都执行一次解压缩处理，从而提高了数据写入性能。

[49]在一些实施例中，所述源表和所述目标表分别为不同的命名空间中的数据库表，且所述源表所在的命名空间与所述目标表所在的命名空间相互隔离。可选地，可以为每个命名空间设置一个标志位，其中，具有不同标志位的各个命名空间彼此隔离，具有相同标志位的命名空间彼此不隔离。标志位的长度可以基于命名空间的数量确定。例如，假设命名空间的数量为 2，则可以采用 1 比特的标志位，例如，“0”和“1”。在实际应用中，源表所在的命名空间可以是对用户可见的命名空间，目标表所在的命名空间可以是对用户不可见的命名空间，即，目标表所在的命名空间中的表为不可见状态。通过不同的标志位可以表征一个命名空间的可见状态。例如，用标志位“0”表示对用户不可见，用标志位“1”表示对用户可见。

[50]通过将源表与目标表设置在不同的命名空间中，一方面，能够使数据重整过程在用户无感知的情况下进行；另一方面，目标表不会占用源表的命名空间中的名称，即，源表所在的命名空间中可以包括与目标表所在的命名空间中具有相同命名方式的数据库表。

[51]在所述源表和所述目标表分别为不同的命名空间中的数据库表的情况下，在执行步骤 106 之后，还可以将所述目标表移动到所述源表所在的命名空间。具体来说，可以将所述目标表的名称更改为所述源表的名称，还可以将所述目标表所在的命名空间的标志位更改为与所述源表所在的命名空间的标志位相同的标志位。

[52]其中，为了方便处理，在更改目标表的名称时，可以直接将目标表的名称与源表的名称互换。可以理解，名称互换仅为一种可选的实现方式。在其他实现方式中，也可以

将目标表的名称更改为源表的名称，并将源表的名称更改为其他名称，以防止目标表的名称与源表的名称重复。例如，假设更改名称之前源表的名称为 x，目标表的名称为 y，则可以将目标表的名称与源表的名称互换，互换后源表的名称为 y，目标表的名称为 x。或者，可以将目标表的名称更改为 x，并将源表的名称更改为 z。进一步地，在将所述目标表的名称更改为所述源表的名称之前，可以为目标表和源表分别设置写锁。在所述源表和所述目标表均设置成功之后，可以将所述目标表的名称更改为所述源表的名称。

[53]在将所述目标表移动到所述源表所在的命名空间之后，还可以删除所述源表。

[54]在一些实施例中，所述目标表包括多个目标表副本，所述多个目标表副本位于分布式数据库中的多个数据库节点上，参见图 4，分布式数据库的多个数据库节点包括主节点和至少一个备份节点。图中以包括 3 个数据库节点的分布式数据库为例进行说明，其中，主节点为节点 2，备用节点为节点 1 和节点 3。可以理解，以上仅为示例性说明，在其他实施例中，分布式数据库中数据库节点的数量不限于此。

[55]为了实现冗余备份等目的，多个数据库节点可以包括相同的目标表副本。在将排序后的数据按数据块写入目标表所在的数据库节点的磁盘时，需要将排序后的数据按数据块分别同步到各个目标表副本所在的数据库节点的磁盘，以保持各个目标表副本的一致性。

[56]在一些实施例中，可以将排序后的数据按数据块写入所述主节点的目标表副本，并将写入所述主节点的目标表副本的数据块写入所述主节点的重做日志（redo log）。然后，将所述主节点的重做日志同步到各个备份节点的重做日志，以使各个备份节点基于本节点的重做日志将排序后的数据按数据块写入本节点的磁盘。

[57]其中，重做日志用于对数据库表的每笔写操作进行记录，数据库通过回放重做日志，可以构建出完整的数据。本公开实施例通过重做日志实现各个备用节点的目标表副本与主节点的目标表副本之间的数据同步，同步过程无需重走事务逻辑，因此，重做日志中不需要记录事务的版本号等信息，减少了重做日志中的数据量，从而减少了目标表副本同步过程中的带宽占用。此外，通过重做日志进行数据同步可以复用已有的重做日志的传输能力，提高开发效率。

[58]在一些实施例中，可以以微块为单位向主节点的重做日志写入数据块，并将主节点的重做日志同步到备用节点。在备用节点的重做日志中的数据块累积到一个宏块的大小时，将备用节点中的宏块写入备用节点的磁盘。

[59]上述实施例中介绍了将源表中的数据迁移到目标表的方式，除此之外，还可以将源表中的依赖对象迁移到目标表。其中，依赖对象包括但不限于索引、约束信息、外键和触发器等中的至少一种。

[60]在依赖对象包括索引的情况下，可以生成目标表的索引。例如，可以基于步骤 106 生成的 SSTable 生成目标表的索引。又例如，在目标表包括源表中的增量数据，且该增量数据被写入目标表所在数据库节点的内存表的情况下，可以基于目标表的 SSTable 和内存表共同生成目标表的索引。

[61]进一步地，在目标表包括多个目标表副本，且所述多个目标表副本位于分布式数据库中的多个数据库节点上的情况下，可以基于所述多个数据库节点中的主节点的目标表的 SSTable 创建所述主节点的目标表副本的索引；将所述目标表副本的索引写入所述主节点的重做日志，并将所述主节点的重做日志同步到所述多个数据库节点中的各个备用节点，以使所述各个备用节点基于本节点的重做日志创建本节点的目标表副本的索引。通过重做日志对索引进行同步的方式与通过重做日志对目标表副本进行同步的方式类似，此处不再赘述。用于对索引进行同步的重做日志与用于对目标表副本进行同步的重做日志可以相同，也可以不同，本公开对此不做限制。

[62]在依赖对象包括约束信息和/或外键等目标项的情况下，可以基于所述目标表的 SSTable 批量读取所述目标表中对应于所述第一目标项的第二目标项，每次读取的第二目标项对应于所述目标表中至少一种类型的数据。其中，第一目标项与对应的第二目标项包括的字段相同。仍以包括订单号、订单金额和下单时间的订单表为例，订单号、订单金额和下单时间互为不同类型的数据。例如，假设源表中的第一目标项为订单号，则目标表中的第二目标项也为订单号。

[63]在批量读取第二目标项时，可以在第一次批量读取时，读取订单号类型的数据，在第二次批量读取时，读取订单金额类型的数据，并在第三次批量读取时，读取下单时间类型的数据。或者，也可以在第一次批量读取时，读取订单号和订单金额两种类型的数据，并在第二次批量读取时，读取下单时间类型的数据。

[64]然后，可以基于读取的第二目标项对应的数据生成校验向量，并对读取的第二目标项中每种类型的数据的校验向量进行校验。一个校验向量中包括的各个元素为同一类型的数据，例如，在一次读取的数据包括订单号类型和订单金额类型的数据时，可以基于订单号类型的数据生成第一校验向量，并基于订单金额类型的数据生成第二校验向量。其中，第一校验向量中包括的各个元素分别为不同订单的订单号，第二校验向量中包括

的各个元素分别为不同订单的订单金额。然后，可以基于第二目标项中对应于第一校验向量的校验逻辑对第一校验向量进行校验，并基于第二目标项中对应于第二校验向量的校验逻辑对第二校验向量进行校验。例如，订单号对应的校验逻辑可以是订单号的长度等于预设长度，订单金额对应的校验逻辑可以是订单金额的数值范围在预设范围内。然后，可以基于订单号的长度对第一校验向量中的各个订单号进行校验，并基于订单金额的数值范围对第二校验向量中的各个订单金额进行校验。以第一校验向量为例，假设第一校验向量中各个订单号的长度均等于预设长度，则校验通过；如果第一校验向量中任意一个订单号的长度不等于预设长度，则该订单号校验不通过。

[65]又例如，假设目标表包括外键，该外键在目标表所引用的表中为主键，则可以基于目标表的外键生成第三校验向量，并基于目标表所引用的表的主键生成第四校验向量。然后，基于第三校验向量与第四校验向量之间的校验逻辑对第三校验向量和第四校验向量进行校验。其中，第三校验向量与第四校验向量之间的校验逻辑可以是第三校验向量与第四校验向量的差集为空集。如果第三校验向量与第四校验向量的差集不是空集，则说明校验不通过；否则，说明校验通过。

[66]上述实施例在生成目标表的 SSTable 之后，直接对目标表的第二目标项进行批量处理，因此，无需按行对第二目标项进行处理，提高了第二目标项的处理效率。可以理解，以上仅为示例性说明，在其他实施例中，可以采用其他的校验逻辑对第二目标项进行校验，且同一种类型的数据对应的校验逻辑的种类可以大于 1。

[67]相比于相关技术中按行对依赖对象进行处理的方式，本实施例通过批量地读取约束信息，并按照数据的类型进行批量校验，一方面批量校验可以利用计算机的局部性原理，增加指令和数据缓存的命中率；另一方面，可以利用向量化技术，提高校验效率，从而提高校验的性能。

[68]在依赖对象包括触发器的情况下，可以在所述目标表中创建触发器。其中，在所述目标表中创建触发器的操作可以与将排序后的数据按数据块写入目标表所在的数据库节点的磁盘之后执行的各种操作包括在相同的事务中。这样能够保证数据重整完成后，触发器才会在目标表上进行触发，避免了在数据重整过程中，触发器在目标表触发带来的性能问题。

[69]例如，可以将所述目标表中创建触发器的操作、为所述源表设置写锁的操作、将所述目标表的名称更改为所述源表的名称的操作，以及将所述目标表所在的命名空间的标志位更改为与所述源表所在的命名空间的标志位相同的标志位的操作包括在相同的

事务中。该事务可以记为：事务 1：{为所述源表设置写锁；将所述目标表的名称更改为所述源表的名称；将所述目标表所在的命名空间的标志位更改为与所述源表所在的命名空间的标志位相同的标志位；在所述目标表中创建触发器}

[70]只有事务 1 中的各个操作均执行完成，事务 1 才能够执行完成。如果事务 1 中的任意一项操作执行失败，则事务 1 执行失败。通过这种方式，能够保证事务的原子性。

[71]本领域技术人员可以理解，在具体实施方式的上述方法中，各步骤的撰写顺序并不意味着严格的执行顺序而对实施过程构成任何限定，各步骤的具体执行顺序应当以其功能和可能的内在逻辑确定。

[72]参见图 5，本公开实施例还提供一种数据库表的数据重整装置，所述装置包括：读取模块 502，用于响应于对源表的 DDL 操作，读取所述源表中的数据；所述 DDL 操作能够改变所述源表的存储结构；记录模块 504，用于对读取的数据进行排序，将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，并记录各个数据块在目标表所在的数据库节点的磁盘的存储地址；生成模块 506，用于在各个数据块均写入完成之后，基于各个数据块的存储地址生成所述目标表的 SSTable，所述目标表的 SSTable 包括所述各个数据块。

[73]在一些实施例中，本公开实施例提供的装置具有的功能或包含的模块可以用于执行上文方法实施例描述的方法，其具体实现可以参照上文方法实施例的描述，为了简洁，这里不再赘述。

[74]本说明书实施例还提供一种计算机设备，其至少包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，其中，处理器执行所述程序时实现前述任一实施例所述的方法。

[75]图 6 示出了本说明书实施例所提供的一种更为具体的计算设备硬件结构示意图，该设备可以包括：处理器 602、存储器 604、输入/输出接口 606、通信接口 608 和总线 610。其中处理器 602、存储器 604、输入/输出接口 606 和通信接口 608 通过总线 610 实现彼此之间在设备内部的通信连接。

[76]处理器 602 可以采用通用的 CPU（Central Processing Unit，中央处理器）、微处理器、服务专用集成电路（Application Specific Integrated Circuit，ASIC）、或者一个或多个集成电路等方式实现，用于执行相关程序，以实现本说明书实施例所提供的技术方案。处理器 602 还可以包括显卡，所述显卡可以是 Nvidia titan X 显卡或者 1080Ti 显卡等。

[77]存储器 604 可以采用 ROM(Read Only Memory, 只读存储器)、RAM(Random Access Memory, 随机存取存储器)、静态存储设备、动态存储设备等形式实现。存储器 604 可以存储操作系统和其他服务程序,在通过软件或者固件来实现本说明书实施例所提供的技术方案时,相关的程序代码保存在存储器 604 中,并由处理器 602 来调用执行。

[78]输入/输出接口 606 用于连接输入/输出模块,以实现信息输入及输出。输入输出/模块可以作为组件配置在设备中(图中未示出),也可以外接于设备以提供相应功能。其中输入设备可以包括键盘、鼠标、触摸屏、麦克风、各类传感器等,输出设备可以包括显示器、扬声器、振动器、指示灯等。

[79]通信接口 608 用于连接通信模块(图中未示出),以实现本设备与其他设备的通信交互。其中通信模块可以通过有线方式(例如 USB、网线等)实现通信,也可以通过无线方式(例如移动网络、WIFI、蓝牙等)实现通信。

[80]总线 610 包括一通路,在设备的各个组件(例如处理器 602、存储器 604、输入/输出接口 606 和通信接口 608)之间传输信息。

[81]需要说明的是,尽管上述设备仅示出了处理器 602、存储器 604、输入/输出接口 606、通信接口 608 以及总线 610,但是在具体实施过程中,该设备还可以包括实现正常运行所必需的其他组件。此外,本领域的技术人员可以理解的是,上述设备中也可以仅包含实现本说明书实施例方案所必需的组件,而不必包含图中所示的全部组件。

[82]本公开实施例还提供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前述任一实施例所述的方法。

[83]计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括,但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带,磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括暂存电脑可读媒体(transitory media),如调制的数据信号和载波。

[84]通过以上的实施方式的描述可知,本领域的技术人员可以清楚地了解到本说明书实

实施例可借助软件加必需的通用硬件平台的方式来实现。基于这样的理解，本说明书实施例的技术方案本质上或者说对相关技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品可以存储在存储介质中，如 ROM/RAM、磁碟、光盘等，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行本说明书各个实施例或者实施例的某些部分所述的方法。

[85]上述实施例阐明的系统、装置、模块或单元，具体可以由计算机芯片或实体实现，或者由具有某种功能的产品来实现。一种典型的实现设备为计算机，计算机的具体形式可以是个人计算机、膝上型计算机、蜂窝电话、图像采集装置电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件收发设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任意几种设备的组合。

[86]本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于装置实施例而言，由于其基本相似于方法实施例，所以描述得比较简单，相关之处参见方法实施例的部分说明即可。以上所描述的装置实施例仅仅是示意性的，其中所述作为分离部件说明的模块可以是或者也可以不是物理上分开的，在实施本说明书实施例方案时可以把各模块的功能在同一个或多个软件和/或硬件中实现。也可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。

[87]以上所述仅是本说明书实施例的具体实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本说明书实施例原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本说明书实施例的保护范围。

### 权利要求书

1、一种数据库表的数据重整方法，所述方法包括：

响应于对源表的 DDL 操作，读取所述源表中的数据；所述 DDL 操作能够改变所述源表的存储结构；

对读取的数据进行排序，将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，并记录各个数据块在目标表所在的数据库节点的磁盘的存储地址；

在各个数据块均写入完成之后，基于各个数据块的存储地址生成所述目标表的 SSTable，所述目标表的 SSTable 包括所述各个数据块。

2、根据权利要求 1 所述的方法，所述源表和所述目标表分别为不同的命名空间中的数据库表，且所述源表所在的命名空间与所述目标表所在的命名空间相互隔离；所述方法还包括：

在基于各个数据块的存储地址生成所述目标表的 SSTable 之后，将所述目标表移动到所述源表所在的命名空间。

3、根据权利要求 2 所述的方法，每个命名空间包括一个标志位，标志位不同的命名空间相互隔离；所述将所述目标表移动到所述源表所在的命名空间，包括：

将所述目标表的名称更改为所述源表的名称；以及

将所述目标表所在的命名空间的标志位更改为与所述源表所在的命名空间的标志位相同的标志位。

4、根据权利要求 2 所述的方法，所述目标表所在的命名空间中的表为不可见状态。

5、根据权利要求 2 所述的方法，所述方法还包括：

在将所述目标表移动到所述源表所在的命名空间之后，删除所述源表。

6、根据权利要求 1 所述的方法，所述目标表包括多个目标表副本，所述多个目标表副本位于分布式数据库中的多个数据库节点上，所述多个数据库节点包括主节点和至少一个备份节点；所述将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，包括：

将排序后的数据按数据块写入所述主节点的目标表副本，并将写入所述主节点的目标表副本的数据块写入所述主节点的重做日志；

将所述主节点的重做日志同步到各个备份节点的重做日志，以使各个备份节点基于本节点的重做日志将排序后的数据按数据块写入本节点的磁盘。

7、根据权利要求 1 所述的方法，所述源表包括多个第一分区，不同的第一分区存储在分布式数据库中不同的第一数据库节点上；所述对读取的数据进行排序，包括：

分别读取各个第一数据库节点中存储的第一分区的数据；

对从各个第一数据库节点读取的第一分区的数据进行排序。

8、根据权利要求 7 所述的方法，所述目标表包括多个第二分区，不同的第二分区存储在分布式数据库中不同的第二数据库节点上；所述将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，包括：

基于排序后的数据所属的第二分区，将排序后的数据按数据块写入目标表在的第二数据库节点的磁盘。

9、根据权利要求 1 所述的方法，所述源表包括第一目标项，所述第一目标项包括所述源表的约束信息和/或外键；所述方法还包括：

基于所述目标表的 SSTable 批量读取所述目标表中对应于所述第一目标项的第二目标项；每次读取的第二目标项对应于所述目标表中至少一种类型的数据；

基于读取的第二目标项对应的数据生成校验向量；一个校验向量中包括的各个元素为同一类型的数据；

对读取的第二目标项中每种类型的数据的校验向量进行校验。

10、根据权利要求 3 所述的方法，所述源表包括触发器；所述将所述目标表的名称更改为所述源表的名称，包括：

为所述源表和所述目标表设置写锁，并在所述源表和所述目标表均设置成功之后将所述目标表的名称更改为所述源表的名称；

所述方法还包括：

在所述目标表中创建触发器；

其中，在所述目标表中创建触发器的操作、为所述源表设置写锁的操作、将所述目标表的名称更改为所述源表的名称的操作，以及将所述目标表所在的命名空间的标志位更改为与所述源表所在的命名空间的标志位相同的标志位的操作包括在相同的事务中。

11、根据权利要求 1 所述的方法，所述目标表包括多个目标表副本，所述多个目标表副本位于分布式数据库中的多个数据库节点上，所述多个数据库节点包括主节点和至少一个备份节点；所述方法还包括：

创建所述主节点的目标表副本的索引；

将所述目标表副本的索引写入所述主节点的重做日志；

将所述主节点的重做日志同步到各个备用节点，以使所述各个备用节点基于本节点的重做日志创建本节点的目标表副本的索引。

12、根据权利要求 1 所述的方法，各个数据块在目标表所在的数据库节点的磁盘的

存储地址记录在所述目标表所在的数据库节点的内存中的数据存储结构中。

13、一种数据库表的数据重整装置，所述装置包括：

读取模块，用于响应于对源表的 DDL 操作，读取所述源表中的数据；所述 DDL 操作能够改变所述源表的存储结构；

记录模块，用于对读取的数据进行排序，将排序后的数据按数据块写入目标表所在的数据库节点的磁盘，并记录各个数据块在目标表所在的数据库节点的磁盘的存储地址；

生成模块，用于在各个数据块均写入完成之后，基于各个数据块的存储地址生成所述目标表的 SSTable，所述目标表的 SSTable 包括所述各个数据块。

14、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权利要求 1 至 12 任意一项所述的方法。

15、一种计算机设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现权利要求 1 至 12 任意一项所述的方法。