一种列式存储方法、设备、计算机程序产品及存储介质

本公开要求于 2024 年 02 月 29 日提交中国专利局、申请号为 202410232858.4、申请名称为“一种列式存储方法、设备、计算机程序产品及存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本公开中。

技术领域

本公开涉及数据库技术领域，尤其涉及一种列式存储方法、设备、计算机程序产品及存储介质。

背景技术

目前，数据库中的数据存储方式通常是行式存储。而在数据分析和查询场景中，列式存储则更有优势，这是由于列式存储是按照列的维度对数据进行组织，因此可在数据分析和查询时有效降低对存储介质的读写次数，从而提高性能。

为此，越来越多的数据库产品支持从行式存储到列式存储的转换。但是，现有的转换方案中通常需要阻塞行式存储的在线服务，在该阻塞状态下完成转换，之后才能恢复行式存储的在线服务。这导致数据库产品在一段时间内无法基于行式存储提供在线服务，而且行式存储的存量数据越多，所需付出的时间代价越大，这给用户带来诸多使用不便。

发明内容

本公开的多个方面提供一种列式存储方法、设备、计算机程序产品及存储介质，用以在不影响按行式存储提供在线服务的前提下，实现列式存储。

本公开实施例提供列式存储方法，包括：

响应于列式存储触发事件，按序从目标数据表中拉取数据行进行列式存储，以产生列存文件及其对应的第一主键索引；

对在拉取期间监测到的所述目标数据表上发生的数据操作进行同步，以产生用于存储增量数据行的增量文件及其对应的第二主键索引；

基于在拉取期间监测到的数据操作，为所述第一主键索引记录所需删除的主键；

响应于拉取结束，将所述第一主键索引和所述第二主键索引合并为目标主键索引，并基于所述所需删除的主键对所述目标主键索引和所述列存文件进行修正。

本公开实施例还提供一种电子设备，包括存储器、处理器和通信组件；

所述存储器用于存储一条或多条计算机指令；

所述处理器与所述存储器和所述通信组件耦合，用于执行所述一条或多条计算机指令，以用于执行前述的列式存储方法。

本公开实施例还提供一种存储计算机程序的计算机可读存储介质，当所述计算机程序被一个或多个处理器执行时，致使所述一个或多个处理器执行前述的列式存储方法。

本公开实施例还提供一种计算机程序产品，包括计算机程序，当所述计算机程序被一个或多个处理器执行时，致使所述一个或多个处理器执行前述的列式存储方法。

在本公开实施例中，在保持按行式存储正常提供在线服务的前提下，可响应于触发事件，按序从目标数据表中拉取数据行进行列式存储，以产生列存文件并按实际拉取到的数据行来为列存文件填充第一主键索引。还可对在拉取期间监测到的目标数据表上发生的数据操作进行同步，以产生用于存在增量数据行的增量文件及其对应的第二主键索引，这可保证列式存储在增量方面能够与行式存储保持一致性。以及还可基于拉取期间监测到的数据操作，为第一主键索引记录所需删除的主键；响应于拉取结束，将第一主键索引和第二

主键索引合并为目标主键索引，并基于所需删除的主键对目标主键索引和列存文件进行修正，这样，可通过记录的所需删除的主键将拉取期间发生的存量方面的数据操作复现至目标主键索引和列存文件上，以保证列式存储在存量方面也能够与行式存储达到一致性。

附图说明

此处所说明的附图用来提供对本公开的进一步理解，构成本公开的一部分，本公开的示意性实施例及其说明用于解释本公开，并不构成对本公开的不当限定。在附图中：

图 1 为本公开一示例性实施例提供的一种列式存储方法的流程示意图；

图 2 为本公开一示例性实施例提供的一种列式存储方法的逻辑示意图；

图 3 为本公开一示例性实施例提供的一种示例性数据库系统的架构示意图；

图 4 为本公开一示例性实施例提供的一种列存文件的内部存储结构示意图；

图 5 为本公开提供的对列存文件进行修正的一种示例性处理方案示意图；

图 6 为本公开另一示例性实施例提供的一种电子设备的结构示意图。

具体实施方式

为使本公开的目的、技术方案和优点更加清楚，下面将结合本公开具体实施例及相应的附图对本公开技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本公开一部分实施例，而不是全部的实施例。基于本公开中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本公开保护的范围。

在对本公开各实施例提供的技术方案进行详细说明之前，先对本公开中涉及到的几个技术概念进行简单解释如下。

行式存储，以行为单位进行数据存储，将一整行数据作为一个存储单元。

列式存储，以列为单位进行数据存储。在行式存储转换为列式存储过程中，可将一组数据行存储为一个列存文件，在列存文件中以列为单位进行数据存储，也即是，列存文件中可将一列数据作为一个存储单元（可称为列段 column segment）。

主键，也称为主关键字（primary key），是数据表中的一个或多个字段，它的值用于唯一的标识数据表中的某一条记录（也即是用于标识数据表中的数据行）。

正如背景技术中介绍的那样，目前，从行式存储到列式存储的转换过程中，需要阻塞行式存储的在线服务，才能保证列式存储与行式存储之间的数据一致性。但是，这种需要暂停在线服务的转换方案，给用户带来诸多使用不便。

为此，本公开实施例提出一种列式存储方案，可在不影响行式存储所提供的在线服务的前提下，保证列式存储与行式存储之间的数据一致性。

其中，数据一致性可理解为列式存储的数据状态能够与行式存储的数据状态一致。应当理解的是，列式存储通常滞后于行式存储，因此，本公开实施例的目标是保证列式存储方案执行完毕后所获得的数据状态，能够与行式存储的某一历史时刻下的数据状态一致。

以下结合附图，详细说明本公开各实施例提供的技术方案。

图 1 为本公开一示例性实施例提供的一种列式存储方法的流程示意图，图 2 为本公开一示例性实施例提供的一种列式存储方法的逻辑示意图。该方法可由数据处理装置执行，该数据处理装置可实现为软件、硬件或软件与硬件的结合，该数据处理装置可集成在电子设备中。参考图 1，该方法可包括：

步骤 100、响应于列式存储触发事件，按序从目标数据表中拉取数据行进行列式存储，以产生列存文件及其对应的第一主键索引；

步骤 101、对在拉取期间监测到的目标数据表上发生的数据操作进行同步，以产生用于存储增量数据行的增量文件及其对应的第二主键索引；

步骤 102、基于在拉取期间监测到的数据操作，为第一主键索引记录所需删除的主键；

步骤 103、响应于拉取结束，将第一主键索引和第二主键索引合并为目标主键索引，并基于所需删除的主键对目标主键索引和列存文件进行修正。

本实施例提供的列式存储方法可适用于各种架构的数据库，图 3 为本公开一示例性实施例提供的一种示例性数据库系统的架构示意图。参考图 3，本实施例提供的列式存储方法可适用于分布式数据库系统，分布式数据库系统中可包括一个或多个计算节点以及一个或多个数据节点，其中，数据节点用于存储数据表，而计算节点则可用于处理针对数据表的各种处理事务，例如，联机事务处理（Online Transaction Processing，OLTP）场景中的各种数据操作以及联机分析处理（Online Analytical Processing，OLAP）场景中的各种查询及分析操作等。在这种示例性数据库系统中，本实施例提供的列式存储方法可将数据节点上按照行式存储的数据表转存为按照列式存储的数据表。值得说明的是，本实施例中并不会干扰行式存储的数据表，也不会删除行式存储的数据表，而是为行式存储的数据表生成了一份按照列式存储的副本。实际应用中，各种数据操作依然发生在按照行式存储的数据表上，而本实施例中所产生的按照列式存储的数据表则可用于支持前述的 OLAP 场景，以提升查询及分析性能。也即是，本实施例中支持行式存储和列式存储共存，也能够保证列式存储与行式存储之间的数据一致性。

本实施例中不限定列式存储方法的执行主体。在一些可能的技术构思中，可在数据库系统中增加独立的电子设备来作为本实施例中的执行主体，例如，在图 3 所示的分布式数据库系统中可增加独立节点，以实施列式存储方法。在另一些可能的技术构思中，也可将数据库系统中现有的节点作为本实施例中的执行主体，例如，在图 3 所示的分布式数据库系统中，可在各个数据节点上分布安装本实施例提供的数据处理装置，以实施列式存储方法，这样，数据节点可对自身行式存储的数据表进行列式存储。

另外，本实施例中不限定行式存储下的数据表类型，本实施例中支持对普通表、分区表及分桶表等各种类型的数据表进行列式存储。为便于表述，本实施例中将以行式存储的目标数据表为例，进行技术方案的说明。应当理解的是，目标数据表可以是任一普通表、分区表中的任一分区或者分桶表中的任一分桶等。目标数据表的规格与行式存储下的数据表类型相适配。

参考图 1，在步骤 100 中，可响应于列式存储触发事件，按序从目标数据表中拉取数据行进行列式存储，以产生列存文件及其对应的第一主键索引。

正如前文提及地，本实施例中行式存储可正常提供在线服务，也即是，在列式存储触发事件发生后，目标数据表上可正常发生数据操作，这些数据操作也将在目标数据表上正常生效。其中，列式存储触发事件可以是接收到列式存储启动指令或者是数据库系统中的日志节点开始推送操作日志等事件。这里提到的日志节点可用于以事件形式记录行式存储的数据表上发生的所有数据操作，日志节点中产生的操作日志可以是一种二进制格式的文件（binlog）。

发明人在研究过程中发现，目标数据表上发生的数据操作通常可包括插入操作、删除操作和更新操作：

插入操作（insert），用于在目标数据表上插入一个或多个数据行。

删除操作（delete），用于在目标数据表上删除一个或多个数据行。

更新操作（update），用于在目标数据表上更新一个或多个数据行。值得说明的是，更新操作可拆解为删除操作加插入操作，也即是，更新操作可理解为从目标数据表上先删除数据行再插入数据行。可知，数据操作将导致目标数据表上的数据行发生增量或删除。

实际应用中，目标数据表中的数据行通常是非常多的，因此，步骤 100 中的拉取过程需要一定的耗时才能完成，这导致在拉取期间，目标数据表上可能会发生数据操作。也即是，在拉取期间，目标数据表是动态变化的。对此，本实施例中提出，在步骤 100 中，在列式存储启动后，按序从目标数据表中拉取数据，这种拉取方案无需考虑目标数据表中的

变化情况，而是按照目标数据表中实际的数据状态，按部就班地进行拉取即可。实际应用中，可对目标数据表进行分页查询，每次拉取一个分页，单个分页内可包含若干数据行。在步骤 100 中，在拉取到数据行之后即可进行列式存储，以产生列存文件。这样，在步骤 100 中，随着拉取操作的进行，将逐步为目标数据表产生多个列存文件。可选地，本实施例中的列存文件可采用 ORC 等格式，在此不做限定。

举例来说，在步骤 100 中，在列式存储启动后，即可开始从目标数据表中拉取第一个分页，对应的是第 1-10000 个数据行，拉取到该分页后，即可对其内的数据行进行列式存储，以获得列存文件 1；拉取完上一分页后即可继续拉取第二个分页，对应的是第 10001-20000 个数据行，同样，在拉取到第二个分页后，可获得列存文件 2。这样按序拉取，将逐步为目标数据表产生多个列存文件。

另外，在步骤 100 中，在对拉取到的数据行进行列式存储后，可为数据行构建对应的主键索引项，以填充至第一主键索引中。本实施例中的主键索引用于记录主键与数据位置之间的映射关系，每条映射关系即记录为一个主键索引项。而且，本实施例中提出了两个主键索引，这里先介绍第一主键索引，关于第二主键索引将在后文中进行介绍。本实施例中的第一主键索引用于记录步骤 100 中拉取到的数据行的主键与其在列存文件中的数据位置。承接前文中为目标数据表产生的多个列存文件，一种实例性的数据位置表征结构可以是【列存文件标识，文件中的数据位置】。图 4 为本公开一示例性实施例提供的一种列存文件的内部存储结构示意图，参考图 4，列存文件中包含多个列段 column segment，列存文件中的一个数据位置可指向各个列段中属于同一数据行的所有元素（图 4 中的多个黑色方块）。举例来说，对于主键为 name 的数据行，若其存储在图 4 中的列存文件（标识为 file1）中的数据位置 pos1 上，则该数据行在第一主键索引中的主键索引项可被记录为：name—》【file1，pos1】。

发明人在研究过程中发现，本实施例中提出的这种拉取方案，在拉取结束时，列式存储的数据行相较于目标数据表来说是冗余的。这是由于，在拉取期间目标数据表上的插入操作可能发生在相应数据行被拉取之前，这种情况，插入的数据行将在步骤 100 中被拉取到；另外，在拉取期间目标数据表上的删除操作可能发生在相应数据行被拉取之后，这种情况下，该数据行将在步骤 100 中被拉取到，但是在目标数据表中则已经删除该数据行。相应地，在拉取结束中产生的第一主键索引相较于目标数据表来说也将是冗余的。

针对这种冗余问题，本实施例中提出在拉取期间不做处理，而是在拉取结束后再做修正。

正如前文提及的，本实施例提供的列式存储方法的目标是需要保证列式存储与行式存储之间的数据一致性，而步骤 100 中拉取过程中产生的列存文件及第一主键索引尚无法满足数据一致性，因此，在拉取期间产生的列存文件及第一主键索引不会交付使用，而是先作为后台处理结果进行存储，等待后续合适的时机才会进行交付使用。这里的交付使用是指允许作为前述 OLAP 场景中的查询及分析操作的数据基础。

对此，参考图 2，本实施例中提出，采用独立线程/进程来执行步骤 100 中的拉取操作。为便于说明，图 2 及后文中将该线程/进程描述为存量处理模块。参考图 2，存量处理模块可独立执行，以逐步产生目标数据表对应的列存文件及第一主键索引。

继续参考图 1，在步骤 101 中，可对在拉取期间监测到的目标数据表上发生的数据操作进行同步，以产生用于存储增量数据行的增量文件及其对应的第二主键索引。正如前文提及的，本实施例中不会影响行式存储所提供的在线服务，但是本实施例中提出，在列式存储启动后，会监测拉取期间在目标数据表上发生的数据操作。

在实际应用方案中，可订阅前文提及的日志节点所提供的操作日志，基于此，在列式存储启动后，日志节点可在目标数据表上产生操作日志时主动向本实施例的执行主体推送操作日志，这样，本实施例中可及时地且全面地监测到在拉取期间目标数据表上发生的数据操作。应当理解的是，这些数据操作是影响列式存储与行式存储之间数据一致性的诱因。

参考图 2，本实施例中提出，采用独立线程/进程来处理监测及处理这些数据操作。为便于说明，图 2 及后文中将该线程/进程描述为增量处理模块。可以理解地是，参考图 2，本实施例中提出相互独立的增量处理模块和存量处理模块。另外，参考图 2，我们将从列式存储启动到拉取结束时刻之间的阶段，描述为第一阶段。在第一阶段中，增量处理模块和存量处理模块可并行执行，互不干扰。

以下将对增量处理模块在第一阶段中的处理逻辑进行说明。

参考图 1 和图 2，在步骤 101 中，增量处理模块可对在拉取期间监测到的目标数据表上发生的数据操作进行同步。这里的同步可理解为将行式存储侧发生的数据操作复现在列式存储侧。发明人在研究过程中发现，目标数据表上发生的数据操作可能会指向目标数据表中的任一数据行，但是，在第一阶段中，存量处理模块尚未完成数据行的拉取，因此，增量处理模块并不具备针对目标数据表的全局视角，这导致在拉取期间监测到的目标数据表上的发生的部分数据操作无法直接复现在列式存储侧。

为此，本实施例中提出，将监测到的数据操作分为两类：针对增量的和针对非增量的。其中，针对增量的数据操作可包括在拉取期间导致目标数据表中出现增量数据行的数据操作以及针对增量数据行所执行的数据操作。这里的增量数据行是指在拉取期间目标数据表上插入过的数据行。而针对非增量的数据操作则可包括针对目标数据表中在列式存储启动时已经存在的存量数据行的数据操作。

基于此，在步骤 101 中，在对监测到的针对增量的数据操作进行同步时，对于所新生的增量数据行可存储至增量文件中，这样，对于增量处理模块来说，其所产生的增量文件中可包含拉取期间目标数据表上出现过的所有增量数据行，这使得增量处理模块具备了关于增量数据行的全局视角，因此，可无障碍地将针对增量数据行的数据操作复现在增量文件上。其中，在步骤 101 中，提出了第二主键索引，以用于记录增量数据行的主键与数据位置之间的映射关系。基于第二主键索引可支持为针对增量的数据操作准确查找到在增量文件中的数据位置，从而支持将这类数据操作复现在增量文件上。可以理解的是，第二主键索引中，为增量数据行记录的数据位置是位于本实施例中提出的增量文件中的。而回忆前文可知，第一主键索引中为从目标数据表中拉取到的数据行所记录的数据位置则是位于本实施例中提出的列存文件中的。也即是，本实施例中列式存储侧的数据行是分散存储在增量文件和列存文件两种文件中的。

在步骤 101 中，在对监测到的针对非增量的数据操作进行同步时，由于无法在增量文件及第二主键索引中找到操作对象，因此，这类数据操作并无法直接复现在增量文件上。在步骤 101 中，可认为这类数据操作同步失败。但是发明人在研究过程中发现，这类数据操作对目标数据表的数据状态的影响主要涉及到数据行的删除，这是由于关于目标数据表中数据行的插入（也即增量数据行）已经在步骤 101 中通过增量文件和第二主键索引得以准确复现了，因此，若能通过其它方式来保证准确复现目标数据表上涉及到的数据行的删除，在步骤 101 中这类数据操作同步失败就不会影响列式存储与行式存储之间的数据一致性。

对此，参考图 1 和图 2，本实施例提出，在步骤 102 中，可基于在拉取期间监测到的数据操作，为第一主键索引记录所需删除的主键。

优选地，为了保证后续能够在列式存储侧（具体可以是指在拉取结束时获得的列式文件及第一主键索引上）全面地复现目标数据表在拉取期间所涉及到的数据行删除，可将在拉取期间监测到的目标数据表上发生的删除操作或更新操作所指向的主键，均记录为第一主键索引中所需删除的主键。发明人在研究过程中发现，这种全面记录删除的方式可能导致产生的所需删除的主键中存在未包含在第一主键索引中的主键（例如，某个增量数据行未被拉取到，其主键不会包含在第一主键索引中，但若监测到该增量数据行发生了删除，也会被记录为所需删除的主键），这虽然会导致一些无效的删除动作，但是却可有效保证对拉取期间目标数据表上的数据行删除进行准确且全面地复现，避免在列式存储侧出现漏

删问题。

参考图 2，在第一阶段中，通过前述步骤 100-步骤 102 之间的相互配合，在第一阶段结束时，列式存储与行式存储之间的数据状态可达到一致。此时，列式存储的数据状态是由：列存文件及其对应的第一主键索引，增量文件及其对应的主键索引，以及为第一主键索引记录的所需删除的主键，这几种处理结果来共同体现的。但应当理解的是，在第一阶段结束时，这几种处理结果所共同体现出的数据状态，是与行式存储的数据状态一致的。

在此基础上，参考图 1 和图 2，本实施例中将继续进入第二阶段。第二阶段可理解为从拉取结束到交付使用之间的的阶段。在第二阶段中，增量处理模块和存量处理模块依然可并行执行，互不干扰。

参考图 1，步骤 103 即对应第二阶段，在步骤 103 中，可响应于拉取结束，将第一主键索引和第二主键索引合并为目标主键索引，并基于所需删除的主键对目标主键索引和列存文件进行修正。参考图 2，可由增量处理模块负责执行将第一主键索引和第二主键索引合并为目标主键索引，以及基于所需删除的主键对目标主键索引进行修正这两方面的操作；可由存量处理模块负责执行基于所需删除的主键对列存文件进行标记删除这方面的操作。

对此，在步骤 103 中，存量处理模块可响应于拉取结束，将所产生的第一主键索引提供至增量处理模块，增量处理模块可响应于拉取结束，将其自身所产生的第二主键索引和存量处理模块所提供的第一主键索引和并为目标主键索引。实际应用中，这里的合并无需进行内存的拷贝和移动，而是可通过列表的载体将两种主键索引统一维护。增量处理模块还可基于步骤 101 中为第一主键索引记录的所需删除的主键，对目标主键索引进行修正。这里的修正至少可包括从目标主键索引中将所需删除的主键对应的主键索引项删除。

参考图 2，修正后的目标主键索引可作为第二阶段结束后的其中一项交付结果。

继续参考图 2，在步骤 103 中，增量处理模块可将在步骤 101 中为第一主键索引记录的所需删除的主键提供至存量处理模块，从而可触发存量处理模块基于所需删除的主键对列存文件进行修正。本实施例中，考虑到列存文件中的数据是按列存储的，对列存文件中的数据进行直接删除的成本比较高，为此，优选地，这里的修正可以是对列存文件进行标记删除。其中，一种示例性的标记删除方案可以是：为列存文件构建删除位图（bitmap），通过删除位图可将所需删除的数据行在列存文件中对应的数据位置标记为已删除。这样，为列存文件正确维护删除位图即可表征列存文件中的数据行删除状态，而无需删除列存文件中的真实数据。据此，在步骤 103 中，存量处理模块可基于所需删除的主键对各个列存文件进行标记删除。

参考图 2，修正后的列存文件可作为第二阶段结束后的其中一项交付结果。

另外，值得说明的是，在第二阶段中，增量处理模块仍然保持对监测到的目标数据表上发生的数据操作进行同步，也即是，在第二阶段中，增量文件和目标主键索引中的第二主键索引部分均跟随对数据操作的同步而动态变化，这可保证目标数据表上针对增量的数据操作能够不间断地复现在列式存储侧。其中，本实施例中也可为增量文件维护删除位图（bitmap），这样，在对数据操作进行同步的过程中，需要删除增量数据行时，也可通过对增量文件进行标记删除来实现，而无需删除增量文件中的真实数据。

参考图 2，第二阶段结束时获得的增量文件也可作为其中一项交付结果

据此，参考图 2，在第二阶段结束时，列式存储与行式存储之间的数据状态仍然保持一致。此时，列式存储的数据状态是由：修正后的目标主键索引（其中包含第一主键索引和第二主键索引两部分），修正后的列存文件（包含为其维护的删除位图），以及第二阶段结束时获得的增量文件（包含为其维护的删除位图），这几种交付结果来共同体现的。但应当理解的是，在第二阶段结束时，这几种交付结果所共同体现出的数据状态，是与行式存储的数据状态一致的。

在第二阶段结束后，本实施例中列式存储的上述几种交付结果可交付使用。其中，列

存文件和增量文件，可作为前述 OLAP 场景中的查询及分析操作的数据基础，当然，实际应用中，为了支持前述的 OLAP 场景，可基于列存文件和增量文件生成列存索引（Column Store Index，CSI），而本实施例中提供的目标主键索引则可用于支持将行式存储侧的数据操作同步至列存文件和增量文件上，从而在交付使用后依然保持列式存储与行式存储之间的数据一致性。

具体地，在第二阶段结束后，增量处理模块可继续监测目标数据表上发生的数据操作；若监测到需要插入数据行的数据操作，则可将所需插入的数据行存储至增量文件中并在目标主键索引包含的第二主键索引中添加对应的主键索引项；若监测到需要删除数据行的数据操作，则从目标主键索引中删除对应的主键索引项并对所删除数据行所处的文件（可能是增量文件，或者，可能是列存文件）进行标记删除。

综上，在本实施例中，在保持按行式存储正常提供在线服务的前提下，可响应于触发事件，按序从目标数据表中拉取数据行进行列式存储，以产生列存文件并按实际拉取到的数据行来为列存文件填充第一主键索引。还可对在拉取期间监测到的目标数据表上发生的数据操作进行同步，以产生用于存在增量数据行的增量文件及其对应的第二主键索引，这可保证列式存储在增量方面能够与行式存储保持一致性。以及还可基于拉取期间监测到的数据操作，为第一主键索引记录所需删除的主键；响应于拉取结束，将第一主键索引和第二主键索引合并为目标主键索引，并基于所需删除的主键对目标主键索引和列存文件进行修正，这样，可通过记录的所需删除的主键将拉取期间发生的存量方面的数据操作复现至目标主键索引和列存文件上，以保证列式存储在存量方面也能够与行式存储达到一致性。

在上述或下述实施例中，在第一阶段中，可采用以下处理方案来保证增量文件的正确性：

若在拉取期间监测到目标数据表上发生需要插入数据行的数据操作，则将所插入的数据行作为增量数据行存储至增量文件中；

基于在拉取期间监测到的需要删除增量数据行的数据操作，对增量文件进行标记删除。

其中，需要插入数据行的数据操作可包括前文提及的插入操作（insert）及更新操作（update）等。另外，实际应用中，增量文件可采用 CSV 等格式。出现新的增量数据行时，可在增量文件的尾部进行追加写入操作，以将增量数据行存储至增量文件中。

其中，需要删除增量数据行的数据操作可包括前文提及的删除操作（delete）及更新操作（update）等。如前文提及地，本实施例中不无需删除增量文件中的真实数据，而是通过标记删除的方式来实现在增量文件上复现这些需要删除增量数据行的数据操作。

针对删除操作（delete）及更新操作（update），在拉取期间对增量文件进行标记删除的实现方案可以是：

若在拉取期间监测到目标数据表上发生针对增量数据行的删除操作，则将增量文件中与所需删除数据行对应的数据位置标记为已删除；若在拉取期间监测到目标数据表上发生针对增量数据行的更新操作，则将增量文件中与因更新操作而删除的数据行对应的数据位置标记为已删除。另外，正如前文提及的，更新操作（update）可理解为先删除数据行再插入数据行，在监测到针对增量数据行的更新操作时，可进行两方面的操作：一方面，可将增量文件中与因更新操作而删除的数据行对应的数据位置标记为已删除；另一方面，可将因更新操作所插入的数据行存储至增量文件中。

这样，在为增量文件维护的删除位图（bitmap）中，需要删除的增量数据行所对应的数据位置将被标记为已删除。

正如前文提及的，在第一阶段中，为增量文件维护的第二主键索引也需跟随对数据操作的同步而动态更新。针对各种数据操作，可采用以下处理方案来保证第二主键索引的正确性：

若在拉取期间监测到目标数据表上发生插入操作，则在第二主键索引中添加与所插入

的增量数据行对应的主键索引项；

若在拉取期间监测到目标数据表上发生针对增量数据行的删除操作，则在第二主键索引中对所需删除数据行对应的主键索引项进行删除；

若在拉取期间监测到目标数据表上发生针对增量数据行的更新操作，则在第二主键索引中将所更新数据行对应的主键索引项指向的数据位置，更新为在增量文件中因更新操作所插入的数据行对应的数据位置。

其中，可以理解的是，在前述对增量文件的处理方案中，增量数据行存储至增量文件后，即可为增量数据行确定出其在增量文件中的数据位置以及增量数据行对应的主键，从而可为增量数据行构建出主键索引项并填充至第二主键索引中。另外，正如前文提及的，更新操作（update）可理解为先删除数据行再插入数据行，这导致所更新的数据行在增量文件中的数据位置将发生变化，但数据行的主键并未发生变化，这种情况下，在第二主键索引中将所更新数据行的主键所映射的数据位置进行更新即可。

综上，本实施例中，针对第一阶段中所监测到的各种针对增量的数据操作，提供了对增量文件及第二主键索引的处理方案，从而可有效保证增量文件及第二主键索引的正确性。应当理解的是，本实施例中提供的处理方案同样适用于第二阶段中所监测到各种针对增量的数据操作，以保证增量文件及第二主键索引在第二阶段中的正确性，在此不做重复赘述。

在上述或下述实施例中，在第二阶段中，可采用以下处理方案来保证对目标主键索引的正确修正：

从目标主键索引包含的第一主键索引部分中，将所需删除的主键对应的主键索引项删除。这可保证在第二阶段阶段时，目标主键索引中所存在的主键能够与目标数据表中实际存在的主键达到一致。

发明人在研究过程中发现，部分增量数据行所对应的主键可能重复出现在目标主键索引中的第一主键索引部分和第二主键索引部分中。这是由于，在第一阶段中，目标数据表上的部分增量数据行被拉取到并列式存储至列存文件中，这类增量数据行对应的主键即会出现在第一主键索引部分中，而正如第一阶段中提及的，这类增量数据行也会出现在第二主键索引部分中。而在第一阶段中，这类增量数据行未被删除的话，即会导致目标主键索引中的第一主键索引部分和第二主键索引部分中出现重复的主键，不过，这类重复的主键在第一主键索引部分和第二主键索引部分中所映射的数据位置是不同的，这是由于这类重复的主键所对应的增量数据行同时存储在增量文件和列存文件中。

值得说明的是，这类重复的主键不会影响列式存储的正确性，但是会占用多余的内存空间。为此，本实施例中进一步提出，在第二阶段中，还可从目标主键所应包含的第一主键索引部分中，将与第二主键索引部分中重复的主键所对应的主键索引项删除。

这样，在第二阶段中对目标主键索引的修正将涉及两个方面：

1、从第一主键索引部分中，将所需删除的主键所对应的主键索引项进行删除。

2、从第一主键索引部分中，将与第二主键索引部分中重复的主键所对应的主键索引项进行删除。

经过这两个方面的修正，可保证第二阶段结束时，目标主键索引能够与目标数据表中的主键达到一致。

另外，在第二阶段中，至少可采用以下两种示例性处理方案来保证对列存文件的正确修正。

在第一种示例性处理方案中：若查询到列存文件中任一数据行所对应的主键为所需删除的主键，则将列存文件中与数据行对应的数据位置标记为已删除，以对列存文件进行标记删除。

实际应用中，可遍历列存文件中的各数据位置，以确定哪些数据位置需要标记为已删

除。应当理解的是，列存文件中的每个数据位置用于存储一个数据行，且记录有数据行对应的主键，因此，在遍历过程中，可判断所遍历到的数据位置上记录的主键是否能够在所需删除的主键中查询到，如果能够查询到，则将遍历到的该数据位置标记为已删除。

另外，值得说明的是，在第二阶段中完成对目标主键索引的修复后，增量处理模块即可获得与目标数据表的实际数据状态一致的全局视角，此后增量处理模块可基于目标主键索引，无障碍地将监测到的目标数据表上发生数据操作同步至增量文件或列存文件中。因对数据操作进行同步而导致的列存文件上的标记删除，与基于所需删除的主键而导致的列存文件上的标记删除可并行进行，两个方面互不干扰，不会出现冲突。

在第二种示例性处理方案中：若查询到列存文件中任一数据行所对应的主键不是所需删除的主键，则将数据行重写至新文件中；若在对列存文件进行重写期间监测到目标数据表上发生有数据操作，则响应于列存文件重写完成，按照所监测到的数据操作，对所产生的新文件进行标记删除。

图 5 为本公开提供的对列存文件进行修正的一种示例性处理方案示意图。参考图 5，其中示出的即为第二种示例性处理方案。

发明人在研究过程中发现，部分列存文件中所需删除的数据行比较多，这些所需删除的数据行对于列式存储来说其实是些无用数据，但却占用了大量的存储空间，为此，在第二种示例性处理方案中提出对列存文件中需要保留的数据行进行重写，以产生新文件，这样，可将多个列存文件合并为新文件，且可释放大量的存储空间。其中，仍可通过遍历的方式来确定列存文件中哪些数据位置上的数据行需要删除，哪些数据位置上的数据行需要保留。

发明人在研究过程中还发现，重写操作会导致一定的耗时，因此，在重写期间目标数据表上可能会发生数据操作。在第二种示例性处理方案中，在重写完成之前，原本的列存文件不会失效，而且，增量处理模块会将重写期间监测到的针对非增量的数据操作同步至列存文件上，这导致，在重写期间，列存文件中可能会发生变化（除了记录的所需删除的主键之外，重写期间可能出现更多需要删除的主键），但新文件对这些变化并无感知。

为此，参考图 5，在第二种示例性处理方案中提出，将第二阶段划分为两个环节：第一环节和第二环节。其中，第一环节从拉取结束到对列存文件重写结束；第二环节从对列存文件重写结束到新文件交付使用。

参考图 5，在第一环节中，列存文件保持有效，增量处理模块可将监测到的针对存量的数据操作同步至列存文件上，并更新目标主键索引中与列存文件对应的第一主键索引部分。在第一环节中，存量处理模块可响应于重写操作启动，根据目标主键索引为列存文件中各数据行分别记录原始数据位置。应当理解的是，在第一环节中，增量处理模块可能会因监测到的数据操作而更新目标主键索引中与列存文件中部分数据行对应的主键索引项。例如，在第一环节中，列存文件中的数据位置 1 上的数据行被删除，则在目标主键索引中该数据行对应的主键索引项将被删除，但是，存量处理模块为该数据行所记录的原始数据位置则是数据位置 1。又例如，在第一环节中，前述数据位置 1 上的数据行发生更新，则在目标主键索引中该数据行对应的主键索引项所记录的数据位置已经变成了增量文件中的数据位置 5，但是，存量处理模块为该数据行所记录的原始数据位置则是数据位置 1。可以理解的是，这两种情况下，存量处理模块会将该数据行重写至新文件中，但该数据行实际已经被删除，新文件中应将该数据行对应的数据位置标记为已删除。

为此，在第二环节中，存量处理模块可响应于重写完成，若查询到列存文件中已重写的任一数据行在目标主键索引中对应的数据位置已经不同于其对应的原始数据位置或者在目标主键索引中已经不存在对应的主键索引项，则在新文件中将数据行标记为已删除。这样，可保证在第二环节结束时所交付的新文件所对应的删除位图（bitmap）的正确性。

另外，考虑到重写会导致数据行的数据位置发生变动，参考图 4，在第二种示例性处

理方案中，存量处理模块还可在第一环节中为重写至新文件中的各个数据行分别记录在新文件中的数据位置。基于此，在第二环节中，存量处理模块可响应于重写完成，将新文件中未标记为已删除的剩余数据行在新文件中的数据位置更新至目标主键索引中。此后，可将新文件交付使用，并将已完成重写的列存文件失效。这样，可保证在第二环节结束时所交付的目标主键索引中第一主键索引部分能够正确记录相关主键对应的数据位置。

发明人在研究过程中发现，在第二种示例性处理方案中，存量处理模块也需要对目标主键索引进行写操作，为避免存量处理模块与增量处理模块在目标主键索引上发生写冲突，可为目标主键索引配置独占写锁。参考图 5，在第二环节开始之前，可有增量处理模块持有该独占写锁，而响应于重写完成，存量处理模块可获取到该独占写锁，以将新文件中未标记为已删除的剩余数据行在新文件中的数据位置更新至目标主键索引中。在完成该更新操作后，该独占写锁重新交还至增量处理模块。

可知，在第二种示例性处理方案中，通过将第二阶段划分为两个环节，可有效保证第二阶段结束时所交付的新文件所对应的删除位图（bitmap）的正确性，以及保证目标主键索引中第一主键索引部分所记录的相关主键对应的数据位置的正确性。而且，通过独占写锁，可有效避免目标主键索引上可能出现的写冲突问题，进一步保证目标主键索引的正确性。

需要说明的是，在上述实施例及附图中的描述的一些流程中，包含了按照特定顺序出现的多个操作，但是应该清楚了解，这些操作可以不按照其在本文中出现的顺序来执行或并行执行，操作的序号如 101、102 等，仅仅是用于区分开各个不同的操作，序号本身不代表任何的执行顺序。另外，这些流程可以包括更多或更少的操作，并且这些操作可以按顺序执行或并行执行。需要说明的是，本文中的“第一”、“第二”等描述，是用于区分不同的阶段、主键索引或环节等，不代表先后顺序，也不限定“第一”和“第二”是不同的类型。

图 6 为本公开另一示例性实施例提供的一种电子设备的结构示意图。如图 6 所示，该电子设备包括：存储器 60 和处理器 61。

处理器 61，与存储器 60 耦合，用于执行存储器 60 中的计算机程序，以用于：

响应于列式存储触发事件，按序从目标数据表中拉取数据行进行列式存储，以产生列存文件及其对应的第一主键索引；

对在拉取期间监测到的所述目标数据表上发生的数据操作进行同步，以产生用于存储增量数据行的增量文件及其对应的第二主键索引；

基于在拉取期间监测到的数据操作，为所述第一主键索引记录所需删除的主键；

响应于拉取结束，将所述第一主键索引和所述第二主键索引合并为目标主键索引，并基于所述所需删除的主键对所述目标主键索引和所述列存文件进行修正。

在一可选实施例中，处理器 61 对在拉取期间监测到的所述目标数据表上发生的数据操作进行同步，以产生增量文件时，可具体用于：

若在拉取期间监测到所述目标数据表上发生需要插入数据行的数据操作，则将所插入的数据行作为增量数据行存储至所述增量文件中；

基于在拉取期间监测到的需要删除增量数据行的数据操作，对所述增量文件进行标记删除。

在一可选实施例中，处理器 61 在基于在拉取期间监测到的需要删除增量数据行的数据操作，对所述增量文件进行标记删除时，可具体用于：

若在拉取期间监测到所述目标数据表上发生针对增量数据行的删除操作，则将所述增量文件中与所需删除数据行对应的数据位置标记为已删除；

若在拉取期间监测到所述目标数据表上发生针对增量数据行的更新操作，则将所述增

量文件中与因所述更新操作而删除的数据行对应的数据位置标记为已删除。

在一可选实施例中，处理器 61 对在拉取期间监测到的所述目标数据表上发生的数据操作进行同步，以产生第二主键索引时，可具体用于：

若在拉取期间监测到所述目标数据表上发生插入操作，则在所述第二主键索引中添加与所插入的增量数据行对应的主键索引项；

若在拉取期间监测到所述目标数据表上发生针对增量数据行的删除操作，则在所述第二主键索引中对所需删除数据行对应的主键索引项进行删除；

若在拉取期间监测到所述目标数据表上发生针对增量数据行的更新操作，则在所述第二主键索引中将所更新数据行对应的主键索引项指向的数据位置，更新为在所述增量文件中因所述更新操作所插入的数据行对应的数据位置。

在一可选实施例中，处理器 61 在基于在拉取期间监测到的数据操作，为所述第一主键索引记录所需删除的主键时，可具体用于：

将在拉取期间监测到的所述目标数据表上发生的删除操作或更新操作所指向的主键，记录为所需删除的主键。

在一可选实施例中，处理器 61 在基于所述所需删除的主键对所述目标主键索引进行修正时，可具体用于：

从所述目标主键索引中包含的第一主键索引部分中，将所述所需删除的主键对应的主键索引项删除。

在一可选实施例中，处理器 61 还可用于：

从所述目标主键索引包含的第一主键索引部分中，将与所述第二主键索引部分中重复的主键所对应的主键索引项删除。

在一可选实施例中，处理器 61 在基于所述所需删除的主键对所述列存文件进行修正时，可具体用于：

若查询到所述列存文件中任一数据行所对应的主键为所需删除的主键，则将所述列存文件中与所述数据行对应的数据位置标记为已删除，以对所述列存文件进行标记删除。

在一可选实施例中，处理器 61 在基于所述所需删除的主键对所述列存文件进行修正时，可具体用于：

若查询到所述列存文件中任一数据行所对应的主键不是所需删除的主键，则将所述数据行重写至新文件中；

若在对所述列存文件进行重写期间监测到所述目标数据表上发生有数据操作，则响应于所述列存文件重写完成，按照所监测到的所述数据操作，对所产生的新文件进行标记删除。

在一可选实施例中，处理器 61 还可用于：

响应于重写操作启动，根据所述目标主键索引为所述列存文件中各数据行分别记录原始数据位置；

响应于重写完成，若查询到所述列存文件中已重写的任一数据行在所述目标主键索引中对应的数据位置已经不同于其对应的原始数据位置或者在所述目标主键索引中已经不存在对应的主键索引项，则在所述新文件中将所述数据行标记为已删除；

将所述新文件中未标记为已删除的剩余数据行在所述新文件中的数据位置更新至所述目标主键索引中。

在一可选实施例中，可为所述目标主键索引配置独占写锁，在重写完成之前，负责执行重写操作的进程在重写完成后获取所述独占写锁，以将所述新文件中未标记为已删除的剩余数据行在所述新文件中的数据位置更新至所述目标主键索引中。

在一可选实施例中，处理器 61 在完成对所述目标主键索引和所述列存文件进行修正之后，还可用于：

继续监测所述目标数据表上发生的数据操作；

若监测到需要插入数据行的数据操作，则将所需插入的数据行存储至增量文件中并在所述目标主键索引包含的第二主键索引中添加对应的主键索引项；

若监测到需要删除数据行的数据操作，则从所述目标主键索引中删除对应的主键索引项并对所删除数据行所处的文件进行标记删除。

进一步，如图 6 所示，该电子设备还包括：通信组件 62 和电源组件 63 等其它组件。图 6 中仅示意性给出部分组件，并不意味着电子设备只包括图 6 所示组件。

值得说明的是，上述关于电子设备各实施例中的技术细节，可参考前述的方法实施例中的相关描述，为节省篇幅，在此不再赘述，但这不应造成本公开保护范围的损失。

相应地，本公开实施例还提供一种存储有计算机程序的计算机可读存储介质，计算机程序被执行时能够实现上述方法实施例中的各步骤。

相应地，本公开实施例还提供一种计算机程序产品，其所包含的计算机程序被执行时能够实现上述方法实施例中的各步骤。

上述图 6 中的存储器，用于存储计算机程序，并可被配置为存储其它各种数据以支持在计算平台上的操作。这些数据的示例包括用于在计算平台上操作的任何应用程序或方法的指令，联系人数据，电话簿数据，消息，图片，视频等。存储器可以由任何类型的易失性或非易失性存储设备或者它们的组合实现，如静态随机存取存储器（SRAM），电可擦除可编程只读存储器（EEPROM），可擦除可编程只读存储器（EPROM），可编程只读存储器（PROM），只读存储器（ROM），磁存储器，快闪存储器，磁盘或光盘。

上述图 6 中的通信组件，被配置为便于通信组件所在设备和其他设备之间有线或无线方式的通信。通信组件所在设备可以接入基于通信标准的无线网络，如 WiFi，2G、3G、4G/LTE、5G 等移动通信网络，或它们的组合。在一个示例性实施例中，通信组件经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中，所述通信组件还包括近场通信(NFC)模块，以促进短程通信。例如，在 NFC 模块可基于射频识别(RFID)技术，红外数据协会（IrDA）技术，超宽带（UWB）技术，蓝牙（BT）技术和其他技术来实现。

上述图 6 中的电源组件，为电源组件所在设备的各种组件提供电力。电源组件可以包括电源管理系统，一个或多个电源，及其他与为电源组件所在设备生成、管理和分配电力相关联的组件。

本领域内的技术人员应明白，本公开的实施例可提供为方法、系统、或计算机程序产品。因此，本公开可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本公开可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本公开是参照根据本公开实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装

置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

需要说明的是，本公开所涉及的用户信息（包括但不限于用户设备信息、用户个人信息等）和数据（包括但不限于用于分析的数据、存储的数据、展示的数据等），均为经用户授权或者经过各方充分授权的信息和数据，并且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准，并提供有相应的操作入口，供用户选择授权或者拒绝。

以上所述仅为本公开的实施例而已，并不用于限制本公开。对于本领域技术人员来说，本公开可以有各种更改和变化。凡在本公开的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本公开的保护范围之内。

权 利 要 求 书

1、一种列式存储方法，包括：

响应于列式存储触发事件，按序从目标数据表中拉取数据行进行列式存储，以产生列存文件及其对应的第一主键索引；

对在拉取期间监测到的所述目标数据表上发生的数据操作进行同步，以产生用于存储增量数据行的增量文件及其对应的第二主键索引；

基于在拉取期间监测到的数据操作，为所述第一主键索引记录所需删除的主键；

响应于拉取结束，将所述第一主键索引和所述第二主键索引合并为目标主键索引，并基于所述所需删除的主键对所述目标主键索引和所述列存文件进行修正。

2、根据权利要求 1 所述的方法，其中，对在拉取期间监测到的所述目标数据表上发生的数据操作进行同步，以产生增量文件，包括：

若在拉取期间监测到所述目标数据表上发生需要插入数据行的数据操作，则将所插入的数据行作为增量数据行存储至所述增量文件中；

基于在拉取期间监测到的需要删除增量数据行的数据操作，对所述增量文件进行标记删除。

3、根据权利要求 2 所述的方法，其中，基于在拉取期间监测到的需要删除增量数据行的数据操作，对所述增量文件进行标记删除，包括：

若在拉取期间监测到所述目标数据表上发生针对增量数据行的删除操作，则将所述增量文件中与所需删除数据行对应的数据位置标记为已删除；

若在拉取期间监测到所述目标数据表上发生针对增量数据行的更新操作，则将所述增量文件中与因所述更新操作而删除的数据行对应的数据位置标记为已删除。

4、根据权利要求 1-3 任一项所述的方法，其中，对在拉取期间监测到的所述目标数据表上发生的数据操作进行同步，以产生第二主键索引，包括：

若在拉取期间监测到所述目标数据表上发生插入操作，则在所述第二主键索引中添加与所插入的增量数据行对应的主键索引项；

若在拉取期间监测到所述目标数据表上发生针对增量数据行的删除操作，则在所述第二主键索引中对所需删除数据行对应的主键索引项进行删除；

若在拉取期间监测到所述目标数据表上发生针对增量数据行的更新操作，则在所述第二主键索引中将所更新数据行对应的主键索引项指向的数据位置，更新为在所述增量文件中因所述更新操作所插入的数据行对应的数据位置。

5、根据权利要求 1-4 任一项所述的方法，其中，基于在拉取期间监测到的数据操作，为所述第一主键索引记录所需删除的主键，包括：

将在拉取期间监测到的所述目标数据表上发生的删除操作或更新操作所指向的主键，记录为所需删除的主键。

6、根据权利要求 1-5 任一项所述的方法，其中，基于所述所需删除的主键对所述目标主键索引进行修正，包括：

从所述目标主键索引中包含的第一主键索引部分中，将所述所需删除的主键对应的主键索引项删除。

7、根据权利要求 1-6 任一项所述的方法，还包括：

从所述目标主键索引包含的第一主键索引部分中，将与所述第二主键索引部分中重复的主键所对应的主键索引项删除。

8、根据权利要求 1-7 任一项所述的方法，其中，基于所述所需删除的主键对所述列

存文件进行修正，包括：

若查询到所述列存文件中任一数据行所对应的主键为所需删除的主键，则将所述列存文件中与所述数据行对应的数据位置标记为已删除，以对所述列存文件进行标记删除。

9、根据权利要求 1-8 任一项所述的方法，其中，基于所述所需删除的主键对所述列存文件进行修正，包括：

若查询到所述列存文件中任一数据行所对应的主键不是所需删除的主键，则将所述数据行重写至新文件中；

若在对所述列存文件进行重写期间监测到所述目标数据表上发生有数据操作，则响应于所述列存文件重写完成，按照所监测到的所述数据操作，对所产生的新文件进行标记删除。

10、根据权利要求 9 所述的方法，还包括：

响应于重写操作启动，根据所述目标主键索引为所述列存文件中各数据行分别记录原始数据位置；

响应于重写完成，若查询到所述列存文件中已重写的任一数据行在所述目标主键索引中对应的数据位置已经不同于其对应的原始数据位置或者在所述目标主键索引中已经不存在对应的主键索引项，则在所述新文件中将所述数据行标记为已删除；

将所述新文件中未标记为已删除的剩余数据行在所述新文件中的数据位置更新至所述目标主键索引中。

11、根据权利要求 10 所述的方法，其中，为所述目标主键索引配置独占写锁，在重写完成之前，负责执行重写操作的进程在重写完成后获取所述独占写锁，以将所述新文件中未标记为已删除的剩余数据行在所述新文件中的数据位置更新至所述目标主键索引中。

12、根据权利要求 1-11 任一项所述的方法，其中，在完成对所述目标主键索引和所述列存文件进行修正之后，所述方法还包括：

继续监测所述目标数据表上发生的数据操作；

若监测到需要插入数据行的数据操作，则将所需插入的数据行存储至增量文件中并在所述目标主键索引包含的第二主键索引中添加对应的主键索引项；

若监测到需要删除数据行的数据操作，则从所述目标主键索引中删除对应的主键索引项并对所删除数据行所处的文件进行标记删除。

13、一种电子设备，包括存储器、处理器和通信组件；

所述存储器用于存储一条或多条计算机指令；

所述处理器与所述存储器和所述通信组件耦合，用于执行所述一条或多条计算机指令，以用于执行权利要求 1-12 任一项所述的列式存储方法。

14、一种存储计算机程序的计算机可读存储介质，当所述计算机程序被一个或多个处理器执行时，致使所述一个或多个处理器执行权利要求 1-12 任一项所述的列式存储方法。

15、一种计算机程序产品，包括计算机程序，当所述计算机程序被一个或多个处理器执行时，致使所述一个或多个处理器执行权利要求 1-12 任一项所述的列式存储方法。