

说明书

数据处理方法、电子设备及存储介质

5 本申请要求于 2022 年 06 月 02 日提交中国专利局、申请号为 202210626163.5、申请名称为“数据处理方法、电子设备及存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机技术领域，尤其涉及一种数据处理方法、电子设备及存储介质。

10 背景技术

数据库系统普遍使用磁盘存储数据，并使用 page（页，也称为数据页）作为磁盘和内存交换的基本单位。通常，数据页通常存储数据库提供的数据库表中的若干条行记录，通常，维护在数据库中的数据库表的数据量比较大，一个数据库表的全部行记录可能存储在磁盘中的多个数据页中，也即一个数据页存储数据库表中的若干条记录。

15 在一些应用场景中，为满足客户端的数据需求，数据库系统可能需要对数据库表进行随机采样，获取数据库表的随机采样数据。一般做法是：多次对磁盘进行 I/O（Input/Output，输入/输出）访问，将数据库表对应的多个数据页从磁盘加载至内存中；在内存中以全表扫描的方式从多个数据页中读取数据库表中的全部行记录；之后，对所读取的全部行记录进行随机采样，得到数据库表中的随机采样数据。

20 然而，上述对数据库表进行随机采样方式容易出现频繁访问磁盘的情形，消耗磁盘 I/O 资源较多，降低了数据库系统的查询性能，特别是在分布式数据库系统具有海量数据的情况下。

发明内容

25 本申请的多个方面提供一种数据处理方法、电子设备及存储介质，用以减少了对磁盘进行 I/O 访问的次数，节约了磁盘 I/O 资源，增强了数据库系统的查询性能。

本申请实施例提供一种数据处理方法，包括：遍历目标数据库表对应的索引树，以确定索引树的最后一个非叶子层，索引树的叶子层中各个叶子节点关联磁盘中的数据页，不同数据页存储有目标数据库表中的不同行记录；针对最后一个非叶子层上每个非叶子节点，
30 依次从非叶子节点关联的索引页中读取当前索引项，并为当前索引项随机产生一个随机数；若随机数满足随机数据采样条件，则将当前索引项所指向的叶子节点的目标数据页从磁盘中加载至内存中，以及读取并存储内存的目标数据页中的行记录。

本申请实施例还提供一种数据处理方法，包括：接收客户端提交的针对目标数据库表

的 SQL 语句；获取目标数据库表的随机采样数据；将 SQL 语句转化为抽象语法树，以及基于随机采样数据根据抽象语法树生成最终的执行计划；将最终的执行计划转化为物理执行计划，并依据物理执行计划查询目标数据库表所在的数据库，获得查询结果，并向客户端输出查询结果。

5 本申请实施例还提供一种电子设备，包括：存储器和处理器；存储器，用于存储计算机程序；处理器耦合至存储器，用于执行计算机程序以用于执行数据处理方法中的步骤。

本申请实施例还提供一种存储有计算机程序的计算机存储介质，当计算机程序被处理器执行时，致使处理器能够实现数据处理方法中的步骤。

10 在本申请实施例中，采用索引树扫描方式并结合随机采样数据库表的数据页的机制，在获取目标数据库表的随机采样数据的过程中，无需对目标数据库表进行全表扫描，有效地减少了对磁盘进行 I/O 访问的次数，节约了磁盘 I/O 资源，增强了数据库系统的查询性能，特别是在分布式数据库系统具有海量数据的情况下。

附图说明

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

- 图 1 为本申请实施例提供的一种数据库系统的结构示意图；
- 图 2 为示例性的数据页的数据结构；
- 图 3 为示例性的父节点和子节点；
- 20 图 4 为示例性的索引树；
- 图 5 为本申请实施例提供的一种数据处理方法的流程图；
- 图 6 为本申请实施例提供的另一种数据处理方法的流程图；
- 图 7 为示例性的行记录变化情况；
- 图 8 为本申请实施例提供的另一种数据处理方法的流程图；
- 25 图 9 为本申请实施例提供的一种数据处理装置的结构示意图；
- 图 10 为本申请实施例提供的一种电子设备的结构示意图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚，下面将结合本申请具体实施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，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，都属于本申请保护的范围。

30 现有的对数据库表进行随机采样方式容易出现频繁访问磁盘的情形，消耗磁盘 I/O 资源较多，降低了数据库系统的查询性能，特别是在分布式数据库系统具有海量数据的情况下。为此，本申请实施例提供一种数据处理方法、电子设备及存储介质。在本申请实施例

中，采用索引树扫描方式并结合随机采样数据库表的数据页的机制，在获取目标数据库表的随机采样数据的过程中，无需对目标数据库表进行全表扫描，有效地减少了对磁盘进行 I/O 访问的次数，节约了磁盘 I/O 资源，增强了数据库系统的查询性能，特别是在分布式数据库系统具有海量数据的情况下。

5 以下结合附图，详细说明本申请各实施例提供的技术方案。

图 1 为本申请实施例提供的一种数据库系统的结构示意图。参见图 1，数据库系统 (Database System, DBS) 包括数据库服务器和磁盘。数据库服务器承担数据库管理功能，包括系统配置与管理、数据存取与更新管理、数据完整性管理和数据安全性管理；以及承担数据库的查询和修改任务，以及承担数据库的维护功能，但并不限于此。

10 在本实施例中，数据库服务器响应客户端发送的查询请求，查询数据库获取查询结果，并返回给客户端。其中，数据库服务器获取查询结果的一般过程是：数据库服务器访问磁盘将包括查询结果的数据页从磁盘中加载至内存中，以及访问内存从数据页中读取查询结果，并将查询结果返回至客户端。

在本实施例中，数据库系统使用磁盘存储数据，数据库中的各种数据（包括各种数据
15 库表）都是存储在磁盘的磁盘文件里。每个磁盘文件包括多个数据页，数据页是从磁盘中读取数据的最小单位。通常，每个数据库表中的全部行记录（是指数据库表中的一行数据）被存储多个数据页中，每个数据页中包括数据库表中的一个或多个行记录。数据页中每一行数据按照主键大小进行顺序存储，每一行数据都有指针指向下一行数据的位置，组成单向链表，其中，主键唯一标识行数据，其中，数据页中除了存储数据库表中的行记录，还
20 会根据实际应用需求存储其他行数据。参见图 2，大量的数据页按顺序一页一页存放的，两两相邻的数据页之间会采用双向链表的格式互相引用。其中，双向链表中的一个链表指针指向上一个数据页的访问信息，双向链表中的另一个链表指针指向下一个数据页的访问信息。其中，访问信息可以是数据页的页面号，基于数据页的页面号可以搜索到数据页。

在本实施例中，不同于全表扫描方式，采用索引树扫描方式扫描数据库表。其中，索引
25 树的类型例如可以是 Btree 树(也称为 B 树)、B-tree（也称为 B-树）和 B+tree（也称为 B+树），但并不以此为限制。其中，B+tree 具有以下特点：1、非叶子节点只存储键值信息。2、所有叶子节点之间都有链表指针。3、数据记录都存放在叶子节点中。

为了便于理解，以索引树为 B+tree 为例进行说明。确定数据库表的主键(primary key)，主键能够唯一标识数据库表中某一行的属性或属性组。在确定好数据库表的主键后，创建
30 B+tree 索引树形式的主键索引。B+tree 索引树自上而下依次为根节点层、一个或多个非叶子层以及一个叶子层。其中，根节点层包括一个根节点，非叶子层包括一个或多个非叶子节点，叶子层包括一个或多个叶子节点。

在本实施例中，根节点和非叶子节点分别关联一个索引页，索引页维护对数据库表进行索引的索引信息。叶子节点关联一个数据页，数据页包括一个或多个索引树所索引的数据库表中的行记录。也就是说，B+tree 索引树将数据库表中的行记录维护在叶子节点对应
35 数据库表中的行记录。

的数据页中，而除叶子节点之外的其他节点对应的索引页维护索引信息，而不维护数据库中
中行记录。

在本实施例中，索引页以维护索引项的形式维护索引信息。具体而言，一个索引页可
以包括一个或多个索引项，可选的，每个索引项包括一个主键和指向子节点的访问信息。
5 其中，指向子节点的访问信息可以是子节点关联的索引页或数据页的页面号、存储位置信
息或者访问地址信息，但不以此为限。应理解，在非叶子节点为索引树中最后一个非叶子
层中的节点时，该非叶子节点的子节点是叶子节点，则该非叶子节点的索引项包括的是子
节点关联的数据页的访问信息。在非叶子节点为索引树中除最后一个非叶子层之外其他非
叶子层中的节点时，该非叶子节点的子节点也是非叶子节点，则该非叶子节点的索引项包
10 括的是子节点关联的索引页的访问信息。

参见图 3，每个索引项包括主键和子节点关联的索引页或数据页的页面号，索引项所
在的节点为父节点。主键反映数据库表中行记录的唯一行标识，索引项中的主键能够用于
决定是否继续搜索子节点关联的索引页或数据页。具体来说，若待查找的主键小于索引项
中的主键，则说明待查找的主键不在子节点关联的索引页或数据页中。若待查找的主键大
15 于或等于索引项中的主键，则说明待查找的主键存在概率在子节点关联的索引页或数据页
中，继续查找父节点的子节点关联的索引页或数据页。

为了便于理解，假设有一张如表 1 所示的记录（ID，名称）的数据库表，以 ID 为主
键建立如图 4 所示的主键索引形式的 B+tree 索引树。

表 1

ID	名称
1	李某
.....	
100	张某
101	杜某
.....	
200	赵某
.....	
300	白某
301	孙某
.....	
400	车某
401	黄某
.....	
500	路某
501	章某

.....	
600	武某

在图 4 中，根节点对应的索引页的页面号是 100，在页面号为 100 的索引页中记录一条索引项 (401, **102**)。该索引项中 401 是主键，也即是表 1 中的 ID 为 401 的行记录的唯一标识。该索引项中 **102** 是根节点的子节点关联的索引页的页面号。

一个非叶子节点对应的索引页的页面号是 101，在页面号为 101 的索引页中记录三条索引项，分别是 (101, **501**)、(201, **502**)、(301, **503**)。(101, **501**) 索引项中的 101 是主键，也即是表 1 中的 ID 为 101 的行记录的唯一标识，**501** 是该非叶子节点的子节点(是一个叶子节点)关联的数据页的页面号。依次类推，(201, **502**) 索引项和 (301, **503**) 索引项的表征意义，在此不再赘述。

基于索引树，可以进行两种查找运算。一种是在主键的范围查找和分页查找，另一种是从根节点开始，进行随机查找。在本实施例中，在对数据库表进行随机采样时，随机对数据库表对应的任一数据页进行数据采样。

在本实施例中，可以根据各个相邻索引项中的主键进行确定各个索引项的主键范围。具体而言，根据索引页中相邻两个索引项中的主键，确定相邻两个索引项中主键较小的索引项的主键范围。索引页中最后一个索引项的主键范围中的最小值为最后一个索引项中记录的主键，最大值为该索引页所能覆盖的最大值。其中，每个索引页所能覆盖的最大值和该索引页对应的父节点关联。

例如，索引页 101 的三个索引项按照主键从小到大的顺序依次排列分别为索引项(101, 501)、索引项 (201, 502)、索引项 (301, 503)。比较相邻的索引项 (101, 501)、索引项 (201, 502) 中主键，确定主键较小的索引项 (101, 501) 的主键范围为 101 至 200；比较相邻的索引项 (201, 502)、索引项 (301, 503) 中主键，确定主键较小的索引项 (201, 502) 的主键范围为 201 至 300。针对索引项 (301, 503) 的主键范围，基于索引页 101 对应的父节点，确定索引页 101 所能覆盖的最大值为 400，于是，索引项 (301, 503) 的主键范围为 301 至 400。

在本实施例中，可以根据待查找的主键落在哪个索引项对应的主键范围，确定待查找的主键所属的索引项，并遍历该索引项所指向的子节点，以在子节点关联的索引页或数据页中继续查询。

在本实施例中，还可以根据二分法确定待查找的主键所属的索引项。具体而言，针对当前遍历到的索引页，依次将主键较小的若干个索引项中的一个索引项作为当前索引项，将待查找的主键依次与当前索引项进行比较，若待查找的主键大于当前索引项中的主键，则将当前索引项的下一个索引项作为新的当前索引项，重复将待查找的主键依次与当前索引项进行比较的步骤，直至待查找的主键小于或等于当前索引项中的主键。若待查找的主键等于当前索引项中的主键，则待查找的主键所属的索引项即为当前索引项，若待查找的主键所属的索引项小于当前索引项，则待查找的主键所属的索引项即为当前索引项的上一

个索引项。

例如,在页面号的索引页 101 中查找待查找的主键匹配的索引项时,依次将索引项(101, 501)、索引项 (201, 502)、索引项 (301, 503) 作为当前索引项。在当前索引项为索引项 (101, 501) 时,将主键 260 与索引项 (101, 501) 中主键 101 比较,由于主键 260 大于主键 101;这时,继续将主键 260 与索引项 (201, 502) 中主键 101 比较,由于主键 260 大于主键 201;这时,继续将主键 260 与索引项 (301, 503) 中主键 301 比较,由于主键 260 小于主键 301,说明索引项 (301, 503) 的上一个索引项即 (201, 502) 是主键 260 所属的索引项。

实际应用中,所有比根节点或非叶子节点中的第一个索引项中主键小的索引项或者行记录也会保存在一个索引页或数据页中,但是索引页中并不存在指向该索引页或数据页的索引项,为了保存该索引页或数据页的页面号,可以将其保存在其父节点的页面头(或其它预先定义的位置)。后续在遍历索引树搜索较小的主键的时候,依然能定位到相应的数据页中。也就是说,根节点或非叶子节点对应的索引页的第一个子节点的页面号没有对应的索引项指向它,而是存储在索引页的页面头(或者其它预先定义的位置),第一个子节点里面存储的所有索引项或者行记录的键值小于其父节点上第一个索引项里包含的键值。参见图 4,虚线箭头指向的是根节点或非叶子节点对应的索引页的第一个子节点。

例如,在根节点对应的页面号为 100 的索引页的页头中保存的是页面号 101,索引页 101 中存储的是键值小于 401 的主键,主键 401 对应的索引项是页面号为 100 的索引页中的第一个索引项。在第一个非叶子节点对应的页面号为 101 的索引页的页头中保存的是页面号 500。页面号为 500 的数据页存储的是键值小于 101 的主键,主键 101 对应的索引项是页面号为 101 的索引页中的第一个索引项。在第二个非叶子节点对应的页面号为 102 的索引页的页头中保存的是页面号 504。页面号为 504 的数据页存储的是键值小于 501 的主键,主键 501 对应的索引项是页面号为 10,的索引页中的第一个索引项。

以待查找的主键为 90 为例,从根节点开始遍历,由于 90 小于根节点对应的第一个索引项 (401, 100) 中包含的主键 401,所以需要从页面号为 100 的索引页的页头中读取其第一个子节点对应的页面号 101。如图 4 中的虚线箭头从页面号为 100 的索引页指向页面号为 101 的索引页所示,接下来在索引页 101 上进行搜索,由于 90 小于第一个索引项(101, 500) 中包含的主键值 101,所以需要从页面号 101 的索引页的页面头读取其第一个子节点对应的页面号 500,并最终在页面号为 500 的数据页中查找主键 90 对应的行记录。

下面结合图 5,对采用索引树扫描方式扫描数据库表进行详细说明。图 5 为本申请实施例提供的一种数据处理方法的流程图。该方法可由数据处理装置执行,该装置可以由软件和/或硬件的方式实现,并一般可以集成在数据库系统中,例如,集成在数据库服务器中。

参见图 5,该方法可以包括以下步骤:

501、遍历目标数据库表对应的索引树,以确定索引树的最后一个非叶子层,索引

树的叶子层中各个叶子节点关联磁盘中的数据页，不同数据页存储有目标数据库表中的不同行记录。

502、针对最后一个非叶子层上每个非叶子节点，依次从非叶子节点关联的索引页中读取当前索引项，并为当前索引项随机产生一个随机数；

- 5 503、若随机数满足随机数据采样条件，则将当前索引项所指向的叶子节点的目标数据页从磁盘中加载至内存中。

504、读取并存储内存的目标数据页中的行记录。

- 10 在本实施例中，目标数据库表是维护在数据库中的任一数据库表。实际应用中，可能需要对数据库中的多张数据库表进行数据扫描，则将需要数据扫描的各张数据库表分别作为目标数据库表。

- 15 在本实施例中，预先建立对目标数据库表进行索引扫描的索引树。以索引树为主键索引形式的 B+tree 为例，根节点和非叶子节点关联的索引页包括一个或多个索引项，可选的，每个索引项包括一个主键和指向子节点的访问信息。其中，指向子节点的访问信息可以是子节点关联的索引页或数据页的页面号和存储位置信息，但不以此为限。值得注意的是，预先将子节点的关联的索引页或数据页的存储位置信息与其页面号进行了关联。于是，基于页面号同样可以访问到对应的索引页或数据页。其中，存储位置信息例如可以是在磁盘或者内存中的存储位置信息，但并不以此为限。

在本实施例中，对需要进行随机采样的目标数据库表，从索引树的根节点开始遍历各层的节点，直至遍历到索引树的最后一个非叶子层。

- 20 在遍历到最后一个非叶子层后，依次遍历非叶子层中的每个非叶子节点。针对当前遍历到的非叶子节点，依次从非叶子节点关联的索引页中读取一个索引项作为当前索引项，并为当前索引项随机产生一个随机数；若随机数满足随机数据采样条件，则将当前索引项所指向的叶子节点的目标数据页从磁盘中加载至内存中，以及读取并存储内存的目标数据页中的行记录，完成对该目标数据页的数据采集。若随机数不满足随机数据采样条件，则
25 放弃采集当前索引项所指向的叶子节点的目标数据页的数据。

在本实施例中，随机数据采样条件根据实际应用需求灵活设置。例如，随机数据采样条件为随机数小于或等于指定数值。也就是说，随机数小于或等于指定数值，则进行采样。随机数大于指定数值，则放弃采样。

- 30 其中，指定数值的百分数即为设定的采样率。其中，设定的采样率在最小采样率和最大采样率之间选取。最小采样率和最大采样率是根据实际应用需求灵活设置的采样率。举例来说，若目标数据库表有 10000 条行记录，最小采样率为 1%，最大采样率为 99%。也即可以从目标数据库表中最少采样 100 条行记录，以及最多采样 9900 条行记录，设定的采样率可以为 [1%，99%] 数值区间中任一数值，指定数值为 [1，99] 数值区间中任一数值。若设定的采样率为 10%，指定数值记为 10，也即若产生的随机数小于或等于 10，进行采样；
35 若产生的随机数大于 10，放弃采样。

以图 4 为例，在遍历到页面号为 101 的索引页中的(101, 501) 索引项时，当前产生的随机数小于或等于指定数值，则对页面号为 501 的数据页进行数据采集。当前产生的随机数大于指定数值，则放弃对页面号为 501 的数据页进行数据采集。

在本实施例中，针对确定进行数据采集的目标数据页，访问磁盘以将该目标数据页从
5 磁盘加载至内存中，接着，访问内存以读取已经加载的目标数据页中的行记录，并对所读取的行记录进行存储，至此完成对目标数据页的数据采集。在索引树上最后一个非叶子层中全部的非叶子节点完成遍历后，也即完成了对目标数据库表中的数据页进行随机采样的整个任务，这时，所读取的各个目标数据页的行记录即为目标数据库表的随机采样数据。

在本实施例中，针对访问内存读取到的目标数据页中的行记录，可以在指定存储
10 位置进行存储，也可以在专用的缓存池中进行缓存，也可以直接发送给客户端，对此不作限制。当然，也可以在读取并存储内存的目标数据页中的行记录之后，将存储的行记录发送给客户端。

在一些应用场景中，数据库系统需要对维护在数据库中数据(具体是指数据库表中的行记录)进行数据统计，统计信息例如包括但不限于：数据量大小、数据分布、唯一值等统计
15 信息。于是，在一些实施例中，对存储的行记录进行信息统计，将统计结果信息发送给客户端。值得注意的是，可以在完成一张目标数据库表的随机采样任务后，启动对存储的全部行记录进行信息统计。也可以在完成多个目标数据库表的随机采样任务后，启动对存储的全部行记录进行信息统计，具体不作限制。

针对具有 1000000000 条行记录的目标数据库表，若采样率为 50%，则需要从目标
20 数据库表采样出一半的行记录。假定每个数据页的大小为 16KB (kilobyte)，若采用现有的全表扫描方式，消耗的磁盘 I/O 数目为 6250000；若采用本案的索引树扫描方式，消耗的磁盘 I/O 数目为 3128906。可以看到，在理想情况下，本案的索引树扫描方式消耗的磁盘 I/O 数目是全表扫描方式的一半，另外，采样率越小，能省更多的磁盘 I/O，性能优势更明显。

25 本申请实施例提供的技术方案，采用索引树扫描方式并结合随机采样数据库表的数据页的机制，在获取目标数据库表的随机采样数据的过程中，无需对目标数据库表进行全表扫描，有效地减少了对磁盘进行 I/O 访问的次数，节约了磁盘 I/O 资源，增强了数据库系统的查询性能，特别是在分布式数据库系统具有海量数据的情况下。

实际应用中，数据页中存储了若干条行记录，为了准确地从内存中逐条读取数据页的
30 行记录，利用叶子游标控制行记录的读取过程。于是，在一些实施例中，读取并存储内存的目标数据页中的行记录的实现过程可以是：响应调用行记录采集接口触发的行记录采集事件，读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录，并将当前读取的行记录进行存储；将当前读取的行记录保存至叶子游标中；重复上述步骤，直至目标数据页中全部行记录读取完毕。

35 在本实施例中，提供用于返回下一条采样数据的行记录采集接口，例如，行记录采集

接口记为 `InnodbSampler::next()`；还提供用于采样初始化的初始化接口，例如，初始化接口记为 `InnodbSampler::init()`；还提供用于在采样结束后进行资源清理的清理接口，例如，清理接口记为 `InnodbSampler::end()`。其中，上述接口可以是 SDK(Software Development Kit, 软件开发工具包)接口，也可以是 API(Application Programming Interface, 应用程序接口)，
5 对此不做限制。

在本实施例中，在启动对数据页进行数据采样之前，可以调用初始化接口执行采样初始化操作。在启动对数据页进行数据采样后，可以调用行记录采集接口执行逐条获取随机采样数据。在结束对数据页进行数据采样后，可以调用清理接口执行资源清理操作。

在本实施例中，除了调用行记录采集接口触发行记录采集事件，还利用叶子游标保存
10 每次读取的行记录，以便于定位下一条需要读取的行记录。这样，在下一个行记录读取之前，利用叶子游标中最后保存的行记录在数据页中准确定位所读取的上一个行记录，定位出的上一个行记录后面的一条行记录即为当前需要读取的下一个行记录。

值得注意的是，叶子游标除了可以保存每次读取的行记录，还可以保存数据页的日志逻辑序列号(log sequence number, LSN)。若数据页中的数据内容发生修改，则数据页的
15 LSN 发生变化。若数据页中的数据内容未发生修改，则数据页的 LSN 保持不变。进一步可选的，为了准确逐行读取数据页中的行记录，结合数据页的 LSN 决策下一条行记录的读取方式。

基于上述，本申请实施例还提供一种数据处理方法，如图 6 所示，该方法可以包括以下步骤：

20 601、遍历目标数据库表对应的索引树，以确定索引树的最后一个非叶子层，索引树的叶子层中各个叶子节点关联磁盘中的数据页，不同数据页存储有目标数据库表中的不同行记录。

602、针对最后一个非叶子层上每个非叶子节点，依次从非叶子节点关联的索引页中读取当前索引项，并为当前索引项随机产生一个随机数；

25 603、若随机数满足随机数据采样条件，则将当前索引项所指向的叶子节点的目标数据页从磁盘中加载至内存中。

604、响应调用行记录采集接口触发的行记录采集事件，判断目标数据页的日志逻辑序列号 LSN 是否发生变化。

30 605、判断目标数据页的日志逻辑序列号 LSN 是否发生变化，若是，执行步骤 608，若否，执行步骤 606。

606、读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录，并将当前读取的行记录进行存储。

607、将当前读取的行记录保存至叶子游标中。

其中，在 607 步骤之后，返回执行步骤 604，直至目标数据页中全部的行记录读
35 取完毕。

608、判断索引页的 LSN 是否发生变化，若是，执行步骤 610，若否，执行步骤 609。

609、根据非叶子游标中最后保存的索引项，在内存中重新定位目标数据页。

在步骤 609 之后，返回执行步骤 606 及后续步骤。

- 5 610、获取叶子游标中最后保存的行记录对应的第一主键；从根节点开始遍历索引树，直至遍历到目标叶子节点，目标叶子节点对应的数据页中包括主键小于或等于第一主键的至少一条行记录。

611、读取加载至内存的目标叶子节点的数据页中位于至少一条行记录后面的下一条行记录，并将当前读取的行记录进行存储。

- 10 在步骤 611 之后，返回执行步骤 607 及后续步骤。

- 在本申请实施例中，既利用叶子游标保存每次从数据页中读取的行记录，还利用非叶子游标保存每次从索引页中读取的索引项，以便于后续快速定位下一条需要读取的行记录。于是，在一些实施例中，在依次从非叶子节点关联的索引页中读取当前索引项之后，还将当前索引项保存至非叶子游标中；在目标数据页中全部的行记录读取
15 完毕后，从非叶子游标中获取最后保存的索引项；在非叶子节点关联的索引页中搜索在最后保存的索引项之后的索引项，并将在最后保存的索引项之后的索引项作为新的当前索引项，直至完成对非叶子节点关联的索引页中全部索引项的处理。

- 在本实施例中，若目标数据页的 LSN 或者索引页的 LSN 未发生改变，可以会以较小的代价读取下一条返回的行记录，从而减少不必要的磁盘 I/O 开销。若目标数据
20 页的 LSN 和索引页的 LSN 均发生改变，则在主键的范围查找方式快速定位下一条需要读取的行记录。具体而言，获取叶子游标中最后保存的行记录对应的主键作为第一主键；从根节点开始遍历索引树，直至遍历到目标叶子节点，目标叶子节点对应的数据页中包括主键小于或等于第一主键的至少一条行记录。读取加载至内存的目标叶子节点的数据页中位于至少一条行记录后面的下一条行记录。

- 25 值得注意的是，若主键小于或等于第一主键的至少一条行记录有多条，则读取主键最大的行记录后面的下一条行记录。

为了便于理解，参见图 7，假设目标数据库表中的行记录 3 是上一次读取的行记录，则在行记录 3 从内存中读取后，无论包括行记录 3 的数据页发生何种变换，调用 InnodbSampler::next()接口都能正确的返回下一条要采样的行记录 4。

- 30 第一种情况：行记录 3 读取前后，其所在的数据页上的行记录无变化。

- 针对上述情况，首先调用 InnodbSampler::next()从数据页读取行记录 3，并将读取的行记录 3 发送给客户端，以及将行记录 3 保存至叶子游标中。接着，调用 InnodbSampler::next()读取触发读取下一条行记录，在读取取下一条行记录时，根据叶子游标中的行记录 3 在数据页中定位出上一次读取的行记录 3，最后，读取在行记录 3 后面的一条行记录即行记录
35 4，并将读取的行记录 4 发送给客户端，以及将行记录 4 保存至叶子游标中。

第二种情况：行记录 3 读取后，其所在的数据页上的行记录 3 被删除。

针对上述情况，首先调用 `InnodbSampler::next()` 从数据页读取行记录 3，并将读取的行记录 3 发送给客户端，以及将行记录 3 保存至叶子游标中。接着，行记录 3 从数据页中删除了。接着，调用 `InnodbSampler::next()` 读取触发读取下一条行记录，具体的，小于或等于行记录 3 对应的主键范围内遍历索引树，直至在数据页找到行记录 2，由于行记录 3 已经被删除，行记录 2 后面的一条行记录为行记录 4，读取行记录 4 并发送给客户端，以及将行记录 4 保存至叶子游标中。

第三种情况：行记录 3 读取后，其所在的数据页上的行记录 1、2、3 均被删除。

值得注意的是，每个数据页上存储两个特殊的行记录，`infimum` 和 `supreme`。其中，`infimum` 表示当前数据页上的逻辑最小行记录；`supreme` 表示当前数据页上的逻辑最大行记录，这两个行记录仅用于标记搜索的边界，不会返回给客户端。

针对上述情况，首先调用 `InnodbSampler::next()` 从数据页读取行记录 3，并将读取的行记录 3 发送给客户端，以及将行记录 3 保存至叶子游标中。接着，行记录 1、2、3 均从数据页中删除了。接着，调用 `InnodbSampler::next()` 读取触发读取下一条行记录，具体的，小于或等于行记录 3 对应的主键范围内遍历索引树，直至在数据页找到行记录 `infimum`，由于行记录 1、2、3 已经被删除，行记录 `infimum` 后面的一条行记录为行记录 4，读取行记录 4 并发送给客户端，以及将行记录 4 保存至叶子游标中。

在基于第一主键遍历索引树的方式是一种在主键的范围查找方式。作为一种示例，从根节点开始遍历索引树，直至遍历到目标叶子节点，包括：针对当前遍历到的除目标叶子节点之外的其他节点，确定其他节点关联的索引页中与第一主键匹配的索引项；遍历与第一主键匹配的索引项所指向的其他节点的子节点。

值得注意的是，若其他节点关联的索引页中仅仅包括一个索引项，则该索引项即为与第一主键匹配的索引项。若其他节点关联的索引页中包括多个索引项，对各个索引项能够索引的主键范围进行分析，第一主键所属的主键范围对应的索引项即为与该第一主键匹配的索引项。值得注意的是，在确定各个索引项的主键范围时，可以根据各个索引项中的主键进行确定。其中，每个索引项记录的主键是最小主键，根据每个索引项记录的最小主键可以确定各个索引项的主键范围。例如，索引页 101 中索引项 (101, 501) 对应的主键范围为 101 至 200；索引页 101 中索引项 (201, 502) 对应的主键范围为 201 至 300；索引页 101 中索引项 (301, 503) 对应的主键范围为 301 至 400。

以图 4 所示的索引树为例，在从页面号为 501 的数据页中读取 (180, 木某) 这条行记录之后，(180, 木某) 这条行记录被返回给客户端，以及 (180, 木某) 这条行记录保存至叶子游标。接着，在读取 (181, 贾某) 行记录之前，确定页面号为 501 的数据页和页面号为 101 的索引页的 LSN 均发生改变，则在小于等于 181 的主键范围内从根节点开始遍历索引树，由于主键 181 小于根节点对应的索引项 (401, 102) 中

的主键 401，从页面号为 100 的索引页的页头中获取查找小于主键 400 的索引页的页面号 101，这时，基于查找到的索引页的页面号 101，遍历页面号为 101 的索引页。101 的索引页的三个索引项中，与主键 181 匹配的是索引项 (101, 501)，于是，遍历页面号为 501 的数据页。

5 在本实施例中，还可以引入锁定机制来保证数据库中数据的一致性。于是，在一些实施例中，在读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录之前，对目标数据页加锁；在读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录之后，释放目标数据页上的锁。值得注意的是，在目标数据页加锁之后，只允许对目标数据页执行规定好的操作，规定好的操作
10 例如为读取目标数据页中的行记录。

 在一些实施例中，在遍历目标数据库表的索引树过程中，在遍历到父节点时，对父节点对应的索引页进行加锁，例如加共享锁 (Shared Locks, S 锁)；在遍历到父节点的子节点时，对子节点的索引页或数据页加锁，并释放父节点对应的索引页上的锁。

 本申请实施例提供的技术方案，采用索引树扫描方式并结合随机采样数据库表的数据
15 页的机制，在获取目标数据库表的随机采样数据的过程中，无需对目标数据库表进行全表扫描，有效地减少了对磁盘进行 I/O 访问的次数，节约了磁盘 I/O 资源，增强了数据库系统的查询性能，特别是在分布式数据库系统具有海量数据的情况下。进一步的，利用叶子游标和非叶子游标控制行记录的读取过程，能够更加准确地从内存中逐条读取数据页的行记录。
20

 通常，数据库系统执行 SQL 语句的流程大致是：首先，对客户端提交的 SQL (结构化查询语言, Structured Query Language) 语句执行语法检查(syntax check)操作，接着，对 SQL 语句执行语义检查(semantic check)操作；接着，对 SQL 语句执行解析(parse)生成语法树；接着，基于语法树生成初步的执行计划(execution plan)；接着，对初步的执行计划
25 (execution plan)进行优化，得到最终的执行计划；接着，将最终的执行计划转化成物理执行计划，依据物理执行计划查询数据库以获取 SQL 语句的查询结果，并将查询结果返回至客户端。其中，执行计划的好坏直接影响数据库系统的查询性能。执行计划在生成或优化过程中，需要依赖数据库中的诸如数据量大小、数据分布、唯一值等数据信息。目前，普遍采用全表扫描方式在数据库中进行随机采样，以获取生成或优化执行计划所依赖的数据
30 信息。但是，特别是在分布式数据库系统具有海量数据的情况下，上述方式需要消耗较多的磁盘 I/O 数量，降低了数据库系统的查询性能。

 为此，本申请实施例还提供一种数据处理方法，如图 8 所示，该数据处理方法可以包括以下步骤：

801、接收客户端提交的针对目标数据库表的 SQL 语句。

35 802、获取目标数据库表的随机采样数据。

803、将 SQL 语句转化为抽象语法树，以及基于随机采样数据根据抽象语法树生成最终的执行计划；

804、将最终的执行计划转化为物理执行计划，并依据物理执行计划查询目标数据库

表所在的数据库，获得查询结果，并向客户端输出查询结果。

其中，进一步可选的，基于随机采样数据根据抽象语法树生成最终的执行计划，包括：基于随机采样数据根据抽象语法树生成初步的执行计划；基于随机采样数据优化初步的执行计划，得到最终的执行计划。

5 在本实施例中，可以采用前述方法实施例提供的数据处理方法获取目标数据库表的随机采样数据，对此不再赘述。

本申请实施例提供的技术方案，在处理客户端发起的 SQL 语句过程中，采用索引树扫描方式并结合随机采样数据库表的数据页的机制，获取目标数据库表的随机采样数据。由于在获取目标数据库表的随机采样数据过程中，有效地减少了对磁盘进行 I/O
10 访问的次数，节约了磁盘 I/O 资源，增强了数据库系统的查询性能，特别是在分布式数据库系统具有海量数据的情况下。

图 9 为本申请实施例提供的一种数据处理装置的结构示意图。参见图 9，该装置可以包括：遍历模块 91 和处理模块 92。

其中，遍历模块 91，用于遍历目标数据库表对应的索引树，以确定索引树的最后
15 一个非叶子层，索引树的叶子层中各个叶子节点关联磁盘中的数据页，不同数据页存储有目标数据库表中的不同行记录：

处理模块 92，用于针对最后一个非叶子层上每个非叶子节点，依次从非叶子节点关联的索引页中读取当前索引项，并为当前索引项随机产生一个随机数；

处理模块 92，还用于若随机数满足随机数据采样条件，则将当前索引项所指向的
20 叶子节点的目标数据页从磁盘加载至内存中，以及读取并存储内存的目标数据页中的行记录。

进一步可选的，处理模块 92，在读取并存储内存的目标数据页中的行记录之后，还用于：将存储的行记录发送给客户端；或者，对存储的行记录进行信息统计，将统计结果信息发送给客户端。

25 进一步可选的，处理模块 92 读取并存储内存的目标数据页中的行记录时，具体用于：响应调用行记录采集接口触发的行记录采集事件，读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录，并将当前读取的行记录进行存储；将当前读取的行记录保存至叶子游标中；重复上述步骤，直至目标数据页中全部行记录读取完毕。

30 进一步可选的，处理模块 92 读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录之前，还用于：判断目标数据页的日志逻辑序列号 LSN 是否发生变化；若目标数据页的 LSN 未发生变化，则执行读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录的操作。

进一步可选的，处理模块 92 还用于若目标数据页的 LSN 发生变化，则判断索引
35 页的 LSN 是否发生变化；若索引页的 LSN 未发生变化，则读取非叶子游标中最后保

存的索引项：根据非叶子游标中最后保存的索引项，在内存中重新定位目标数据页，并执行读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录的操作。

进一步可选的，处理模块 92 在依次从非叶子节点关联的索引页中读取当前索引项之后，还用于：将当前索引项保存至非叶子游标中；在目标数据页中全部的行记录读取完毕后，从非叶子游标中获取最后保存的索引项；在非叶子节点关联的索引页中搜索在最后保存的索引项之后的索引项，并将在最后保存的索引项之后的索引项作为新的当前索引项，直至完成对非叶子节点关联的索引页中全部索引项的处理。

进一步可选的，处理模块 92 还用于：若索引页的 LSN 发生变化，则获取叶子游标中最后保存的行记录对应的第一主键；从根节点开始遍历索引树，直至遍历到目标叶子节点，目标叶子节点对应的数据页中包括主键小于或等于第一主键的至少一条行记录；读取加载至内存的目标叶子节点的数据页中位于至少一条行记录后面的下一条行记录，并将当前读取的行记录进行存储；返回执行将当前读取的行记录保存至叶子游标中的步骤。

进一步可选的，处理模块 92 从根节点开始遍历索引树，直至遍历到目标叶子节点时，具体用于：针对当前遍历到的除目标叶子节点之外的其他节点，确定其他节点关联的索引页中与第一主键匹配的索引项；遍历与第一主键匹配的索引项所指向的其他节点的子节点。

进一步可选的，处理模块 92 在读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录之前，还用于：对目标数据页加锁；处理模块 92 在读取加载至内存的目标数据页中在叶子游标中最后保存的行记录后面的一条行记录之后，还用于：释放目标数据页上的锁。

进一步可选的，索引树为 B+tree 索引树。

关于图 9 所示的数据处理装置其中各个模块执行操作的具体方式可参见前述方法实施例中的相关描述，在此不再赘述。

关于处理器执行各动作的详细实施过程可参见前述方法实施例或设备实施例中的相关描述，在此不再赘述。

需要说明的是，上述实施例所提供方法的各步骤的执行主体均可以是同一设备，或者，该方法也由不同设备作为执行主体。比如，步骤 501 至步骤 504 的执行主体可以为设备 A；又比如，步骤 501 和 502 的执行主体可以为设备 A，步骤 503 和 504 的执行主体可以为设备 B；等等。

另外，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 501、502 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，是用于区分不同的消息、设备、模块等，不代表先后顺序，也不限定“第一”和“第二”是不同的类型。

图 10 为本申请实施例提供的一种电子设备的结构示意图。如图 10 所示，该电子设备包括：存储器 101 和处理器 102：

5 存储器 101，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 101 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
10 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。

处理器 102，与存储器 101 耦合，用于执行存储器 101 中的计算机程序，以用于：
遍历目标数据库表对应的索引树，以确定索引树的最后一个非叶子层，索引树的叶子
层中各个叶子节点关联磁盘中的数据页，不同数据页存储有目标数据库表中的不同行
15 记录；针对最后一个非叶子层上每个非叶子节点，依次从非叶子节点关联的索引页中
读取当前索引项，并为当前索引项随机产生一个随机数；若随机数满足随机数据采样
条件，则将当前索引项所指向的叶子节点的目标数据页从磁盘加载至内存中，以及
读取并存储内存的目标数据页中的行记录。

或者，处理器 102，与存储器 101 耦合，用于执行存储器 101 中的计算机程序，
20 以用于：接收客户端提交的针对目标数据库表的 SQL 语句；获取目标数据库表的随机
采样数据；将 SQL 语句转化为抽象语法树，以及基于随机采样数据根据抽象语法树生
成最终的执行计划；将最终的执行计划转化为物理执行计划，并依据物理执行计划查
询目标数据库表所在的数据库，获得查询结果，并向客户端输出查询结果。

进一步，如图 10 所示，该电子设备还包括：通信组件 103、显示器 104、电源组
25 件 105、音频组件 106 等其它组件。图 10 中仅示意性给出部分组件，并不意味着电子
设备只包括图 10 所示组件。另外，图 10 中虚线框内的组件为可选组件，而非必选组
件，具体可视电子设备的形态而定。本实施例的电子设备的形态可以为台式电脑、
笔记本电脑、智能手机或 IOT 设备等终端设备，也可以是常规服务器、云服务器或服
务器阵列等服务端设备。若本实施例的电子设备的形态实现为台式电脑、笔记本电脑、智能
30 手机等终端设备，可以包含图 10 中虚线框内的组件；若本实施例的电子设备的形态实现为常
规服务器、云服务器或服务器阵列等服务端设备，则可以不包含图 10 中虚线框内的组
件。

关于处理器执行各动作的详细实施过程可参见前述方法实施例或系统实施例中的
相关描述，在此不再赘述。

35 相应地，本申请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可由电子设备执行的各步骤。

相应地，本申请实施例还提供一种计算机程序产品，包括计算机程序/指令，当计算机程序/指令被处理器执行时，致使处理器能够实现上述方法实施例中可由电子设备执行的各步骤。

- 5 上述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，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。
- 10

- 上述显示器包括屏幕，其屏幕可以包括液晶显示器（LCD）和触摸面板（TP）。如果屏幕包括触摸面板，屏幕可以被实现为触摸屏，以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。触摸传感器可以不仅感测触摸或滑动动作的边界，而且还检测与触摸或滑动操作相关的持续时间和压力。
- 15

 上述电源组件，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。

- 上述音频组件，可被配置为输出和/或输入音频信号。例如，音频组件包括一个麦克风（MIC），当音频组件所在设备处于操作模式，如呼叫模式、记录模式和语音识别模式时，麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器或经由通信组件发送。在一些实施例中，音频组件还包括一个扬声器，用于输出音频信号。
- 20

- 本领域内的技术人员应明白，本申请的实施例可提供为方法、系统、或计算机程序产品。因此，本申请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，本申请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。
- 25

- 本申请是参照根据本申请实施例的方法、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。
- 30

- 35 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。

- 5 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。

在一个典型的配置中，计算设备包括一个或多个处理器 (CPU)、输入/输出接口、网络接口和内存。

- 10 内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (RAM) 和/或非易失性内存等形式，如只读存储器 (ROM) 或闪存(flash RAM)。内存是计算机可读介质的示例。

- 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)、
15 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
20 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。

- 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。
25 在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括要素的过程、方法、商品或者设备中还存在另外的相同要素。

以上仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本申请的权利要求范围之内。

权 利 要 求 书

- 1、一种数据处理方法，其特征在于，包括：
- 遍历目标数据库表对应的索引树，以确定所述索引树的最后一个非叶子层，所述索引树的叶子层中各个叶子节点关联磁盘中的数据页，不同数据页存储有所述目标数据库表中的不同行记录；
- 5 针对所述最后一个非叶子层上每个非叶子节点，依次从所述非叶子节点关联的索引页中读取当前索引项，并为所述当前索引项随机产生一个随机数；
- 若所述随机数满足随机数据采样条件，则将所述当前索引项所指向的叶子节点的目标数据页从磁盘加载至内存中，以及读取并存储所述内存的所述目标数据页中的行记录。
- 10 2、根据权利要求1所述的方法，其特征在于，在读取并存储所述内存的所述目标数据页中的行记录之后，还包括：
- 将存储的行记录发送给客户端；
- 或者，对存储的行记录进行信息统计，将统计结果信息发送给所述客户端。
- 15 3、根据权利要求1所述的方法，其特征在于，读取并存储所述内存的所述目标数据页中的行记录，包括：
- 响应调用行记录采集接口触发的行记录采集事件，读取加载至所述内存的所述目标数据页中在叶子游标中最后保存的行记录后面的一条行记录，并将当前读取的行记录进行存储；
- 20 将当前读取的行记录保存至所述叶子游标中；重复上述步骤，直至所述目标数据页中全部行记录读取完毕。
- 4、根据权利要求3所述的方法，其特征在于，读取加载至所述内存的所述目标数据页中在叶子游标中最后保存的行记录后面的一条行记录之前，还包括：
- 判断所述目标数据页的日志逻辑序列号LSN是否发生变化；
- 25 若所述目标数据页的LSN未发生变化，则执行读取加载至所述内存的所述目标数据页中在叶子游标中最后保存的行记录后面的一条行记录的操作。
- 5、根据权利要求4所述的方法，其特征在于，还包括：
- 若所述目标数据页的LSN发生变化，则判断所述索引页的LSN是否发生变化；
- 若所述索引页的LSN未发生变化，则读取非叶子游标中最后保存的索引项；
- 30 根据所述非叶子游标中最后保存的索引项，在所述内存中重新定位所述目标数据页，并执行读取加载至所述内存的所述目标数据页中在叶子游标中最后保存的行记录后面的一条行记录的操作。
- 6、根据权利要求5所述的方法，其特征在于，在依次从所述非叶子节点关联的索引页中读取当前索引项之后，还包括：

将所述当前索引项保存至非叶子游标中；

在所述目标数据页中全部的行记录读取完毕后，从所述非叶子游标中获取最后保存的索引项；

- 5 在所述非叶子节点关联的索引页中搜索在所述最后保存的索引项之后的索引项，并将在所述最后保存的索引项之后的索引项作为新的当前索引项，直至完成对所述非叶子节点关联的索引页中全部索引项的处理。

7、根据权利要求 5 所述的方法，其特征在于，还包括：

若所述索引页的 LSN 发生变化，则获取所述叶子游标中最后保存的行记录对应的第一主键；

- 10 从根节点开始遍历所述索引树，直至遍历到目标叶子节点，所述目标叶子节点对应的数据页中包括主键小于或等于所述第一主键的至少一条行记录；

读取加载至所述内存的所述目标叶子节点的数据页中位于所述至少一条行记录后面的下一条行记录，并将当前读取的行记录进行存储；

返回执行将当前读取的行记录保存至所述叶子游标中的步骤。

- 15 8、根据权利要求 7 所述的方法，其特征在于，从根节点开始遍历所述索引树，直至遍历到目标叶子节点，包括：

针对当前遍历到的除所述目标叶子节点之外的其他节点，确定所述其他节点关联的索引页中与所述第一主键匹配的索引项；遍历与第一主键匹配的索引项所指向的所述其他节点的子节点。

- 20 9、根据权利要求 2 所述的方法，其特征在于，在读取加载至所述内存的所述目标数据页中在叶子游标中最后保存的行记录后面的一条行记录之前，还包括：

对所述目标数据页加锁；

读取加载至所述内存的所述目标数据页中在叶子游标中最后保存的行记录后面的一条行记录之后，还包括：

- 25 释放所述目标数据页上的锁。

10、根据权利要求 1 至 8 任一项所述的方法，其特征在于，所述索引树为 B+tree 索引树。

11、一种数据处理方法，其特征在于，包括：

接收客户端提交的针对目标数据库表的 SQL 语句；

- 30 按照权利要求 1 至 10 任一项所述的方法获取所述目标数据库表的随机采样数据；

将所述 SQL 语句转化为抽象语法树，以及基于所述随机采样数据根据所述抽象语法树生成最终的执行计划；

将所述最终的执行计划转化为物理执行计划，并依据所述物理执行计划查询所述目标数据库表所在的数据库，获得查询结果，并向所述客户端输出所述查询结果。

- 35 12、根据权利要求 11 所述的方法，其特征在于，所述基于所述随机采样数据根据所述

抽象语法树生成最终的执行计划，包括：

基于所述随机采样数据根据所述抽象语法树生成初步的执行计划；

基于所述随机采样数据优化所述初步的执行计划，得到最终的执行计划。

- 13、一种电子设备，其特征在于，包括：存储器和处理器；所述存储器，用于存储计算机程序；所述处理器耦合至所述存储器，用于执行所述计算机程序以用于执行权利要求 1-11 任一项所述方法中的步骤。
- 5

14、一种存储有计算机程序的计算机存储介质，其特征在于，当所述计算机程序被处理器执行时，致使所述处理器能够实现权利要求 1-11 任一项所述方法中的步骤。