



(12) 发明专利申请

(10) 申请公布号 CN 104111992 A

(43) 申请公布日 2014. 10. 22

(21) 申请号 201410315793. 6

(22) 申请日 2014. 07. 03

(71) 申请人 北京思特奇信息技术股份有限公司
地址 100086 北京市海淀区中关村南大街 6
号中电信息大厦 16 层

(72) 发明人 李静

(74) 专利代理机构 北京轻创知识产权代理有限
公司 11212
代理人 杨立

(51) Int. Cl.

G06F 17/30(2006. 01)

H04L 29/06(2006. 01)

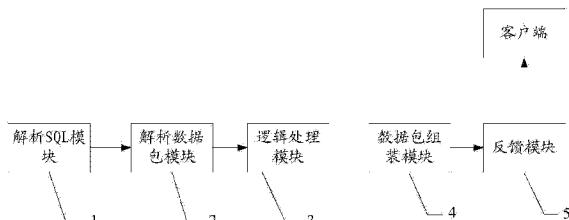
权利要求书4页 说明书9页 附图10页

(54) 发明名称

一种分布式数据库的代理结果合并方法及系
统

(57) 摘要

本发明涉及一种分布式数据库的代理结果合
并方法及系统,其系统包括:解析 SQL 模块、解
析数据包模块、逻辑处理模块、数据包组装模块和反
馈模块;所述解析 SQL 模块用于解析 SQL 语句得
到其包括的函数;所述解析数据包模块用于分析
数据包,提取行数据,列字段数据信息;所述逻辑
处理模块用于提取每行的对应函数所在字段的数
据信息,并进行逻辑处理,根据不同函数进行对
应的处理得到结果;所述数据包组装模块用于将处
理后的结果重新封装成符合 MYSQL 协议的数据
包,并将所述数据包发送到反馈模块;所述反馈
模块用于将数据包发送到客户端。本发明具有经
济性;还能提升性能和安全性。



1. 一种分布式数据库的代理结果合并方法,其特征在于,具体包括以下步骤:

步骤 1 :解析第一 SQL 语句得到其包括的函数;

步骤 2 :判断所述函数属于第一类函数还是第二类函数,如果是第一类函数,执行步骤 8 ;否则,执行步骤 3 ;

步骤 3 :判断所述函数是否是 distinct 函数,如果是,执行步骤 5 ;否则,执行步骤 4 ;

步骤 4 :对第一 SQL 语句进行预处理得到第二 SQL 语句,将第二 SQL 语句替换第一 SQL 语句;

步骤 5 :发送第一 SQL 语句或第二 SQL 语句到 MYSQL 服务端;

步骤 6 :获得 MYSQL 服务端返回的结果集,根据 MYSQL 数据包的协议格式,将结果集转换成数据包集合;

步骤 7 :根据所述函数对数据包集合进行操作,得到新的结果集;

步骤 8 :对新的结果集进行数据包封装得到新的数据包,将新的数据包发送到客户端。

2. 根据权利要求 1 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述第一类函数包括 sum 函数、count 函数、max 函数和 min 函数;所述第二类函数包括 avg 函数、order by 函数、group by 函数和 distinct 函数。

3. 根据权利要求 1 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述步骤 4 具体包括以下步骤:

步骤 4.1 :判断第一 SQL 语句是否包含 avg 函数,如果是,执行步骤 4.2 ;否则执行步骤 4.3 ;

步骤 4.2 :将第一 SQL 语句中的 avg() 改变为 sum(), count() 和 avg(), 构成第二 SQL 语句,执行步骤 5 ;

步骤 4.3 :判断第一 SQL 语句是否包含 order by 函数,如果是,执行步骤 4.4 ;否则,执行步骤 4.6 ;

步骤 4.4 :判断查询字段是否包含 order by 排序字段;如果是,执行步骤 4.6 ;否则,执行步骤 4.5 ;

步骤 4.5 :将第一 SQL 语句中的排序字段添加到查询字段中形成第二 SQL 语句,执行步骤 5 ;

步骤 4.6 :判断第一 SQL 语句是否包含 group by 函数;如果是,执行步骤 4.5 ;否则,执行步骤 5 。

4. 根据权利要求 1-3 任一项所述的一种分布式数据库的代理结果合并方法,其特征在于,所述步骤 7 具体包括以下步骤:

步骤 7.1 :判断所述函数是否是 sum 函数或 count 函数;如果是,执行步骤 7.2 ;否则,执行步骤 7.3 ;

步骤 7.2 :对多节点的结果集进行累加构成新的结果集,执行步骤 8 ;

步骤 7.3 :判断所述函数是否是 max 函数;如果是,执行步骤 7.4 ;否则,执行步骤 7.5 ;

步骤 7.4 :对多节点的结果集进行求最大值得到新的结果集,执行步骤 8 ;

步骤 7.5 :判断所述函数是否是 min 函数;如果是,执行步骤 7.6 ;否则执行步骤 7.7 ;

步骤 7.6 :对多节点的结果集进行求最小值得到新的结果集,执行步骤 8 ;

步骤 7.7 :判断所述函数是否是 avg 函数;如果是,执行步骤 5.8 ;否则,执行步骤 7.9 ;

步骤 7.8 :对所有节点的结果集进行统一求平均值得到新的结果集,执行步骤 8 ;

步骤 7.9 :判断所述函数是否是 order by 函数,如果是,执行步骤 7.10 ;否则,执行步骤 7.11 ;

步骤 7.10 :对多节点的结果集进行重新排序得到新的结果集,执行步骤 8 ;

步骤 7.11 :判断所述函数是否是 group by 函数,如果是,执行步骤 7.12 ;否则,执行步骤 7.13 ;

步骤 7.12 :对多节点的结果集进行重新组装得到新的结果集,执行步骤 8 ;

步骤 7.13 :对单个节点的 distinct 值进行简单合并,得到新的结果集,执行步骤 8 。

5. 根据权利要求 4 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述步骤 7.2 包括以下步骤 :

步骤 7.2.1 :从结果集中获得列字段的列数据包,在列数据包中找到 sum() 字段或 count() 字段,并获得该字段的序列号 ;

步骤 7.2.2 :从结果集中获得行字段的行数据包,在行数据包中,获得每一行的 sum() 函数或 count() 函数的结果,并进行累加得到累加值 ;

步骤 7.2.3 :将累加值放到第一行数据包里,删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8 。

6. 根据权利要求 4 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述所述步骤 7.4 具体包括以下步骤 :

步骤 7.4.1 :从结果集中获得列字段的列数据包,在列数据包中找到 max() 字段,并获得该字段的序列号 ;

步骤 7.4.2 :从结果集中获得行字段的行数据包,在行数据包中查找每一行的 max() 函数的结果,并比较大小获得其最大值 ;

步骤 7.4.3 :将最大值放到第一行数据包里,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8 。

7. 根据权利要求 4 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述步骤 7.6 具体包括以下步骤 :

步骤 7.6.1 :从结果集中获得列字段的列数据包,在列数据包中找到 min() 字段,并获得该字段的序列号 ;

步骤 7.6.2 :从结果集中获得行字段的行数据包,在行数据包中查找每一行的 min() 函数的结果,并比较大小获得其最小值 ;

步骤 7.6.3 :将最小值放到第一行数据包里,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8 。

8. 根据权利要求 4 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述步骤 7.8 具体包括以下步骤 :

步骤 7.8.1 :从结果集中获得列字段的列数据包,在列数据包中找到 avg() 、 sum() 和 count() 字段,并获得该字段的序列号 ;

步骤 7.8.2 :从结果集中获得行字段的行数据包,在行数据包中获得每一行的 sum 函数和 count 函数的结果,并且分别累加得到总的 sum 值和 count 值 ;

步骤 7.8.3 :sum 值除以 count 值得到平均值,并组装成行数据包 ;

步骤 7.8.4 :删除多余行数据包和多余列数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8。

9. 根据权利要求 4 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述步骤 7.10 具体包括以下步骤 :

步骤 7.10.1 :从结果集中获得列字段的列数据包,查找得到包含排序字段的列字段的序列号 ;

步骤 7.10.2 :从结果集中获得行数据包,从行数据包中获得每一行的排序字段的值,并且将排序字段的值保持到结果集中 ;

步骤 7.10.3 :判断所述第二 SQL 语句中是否包含 desc,如果是,执行步骤 7.10.4 ;否则,执行步骤 7.10.5 ;

步骤 7.10.4 :对结果集中的值进行降序排列得到新的结果集,在新的结构集中代表包序列号的信息更改为正确的序列号的信息,执行步骤 8 ;

步骤 7.10.5 :对结果集中的值进行升序排列并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8。

10. 根据权利要求 4 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述步骤 7.12 具体包括以下步骤 :

步骤 7.12.1 :从结果集中获得列字段的列数据包,在列数据包中查找得到包含 group by 字段的列字段的序列号 ;

步骤 7.12.2 :从结果集中获得行字段的行数据包,并获得每一行的排序字段的值,将排序字段的值存入结果集中 ;

步骤 7.12.3 :对结果集中的值进行升序排列,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8。

11. 根据权利要求 4 所述的一种分布式数据库的代理结果合并方法,其特征在于,所述步骤 7.13 具体包括以下步骤 :

步骤 7.13.1 :从结果集中获得列字段的列数据包,在列数据包中查找得到包含 distinct 字段的列字段的序列号 ;

步骤 7.13.2 :从结果集中获得行字段的行数据包,并获得每一行的排序字段的值,将排序字段的值存入结果集中 ;

步骤 7.13.3 :对结果集中的值进行筛选剔除重复的数据,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8。

12. 一种分布式数据库的代理结果合并系统,其特征在于,包括 :解析 SQL 模块、解析数据包模块、逻辑处理模块、数据包组装模块和反馈模块 ;

所述解析 SQL 模块用于解析 SQL 语句得到其包括的函数 ;

所述解析数据包模块用于分析数据包,提取行数据,列字段数据信息 ;

所述逻辑处理模块用于提取每行的对应函数所在字段的数据信息,并进行逻辑处理,根据不同函数进行对应的处理得到结果 ;

所述数据包组装模块用于将处理后的结果重新封装成符合 MYSQL 协议的数据包,并将所述数据包发送到反馈模块 ;

所述反馈模块用于将数据包发送到客户端。

一种分布式数据库的代理结果合并方法及系统

技术领域

[0001] 本发明涉及一种分布式数据库的代理结果合并方法及系统。

背景技术

[0002] 现有分布式数据库代理对客户端发送来待处理的每一个 SQL 请求, 经过代理的路由规则得到目标节点, 然后将 SQL 信息发送给 MySQL 服务器处理, 并将处理结果返回给客户端。如果路由结果是多个物理节点, 那么代理会将查询结果做简单的合并返回给客户端。如果 SQL 里没有一些诸如 sum、max 等函数, 那么查询结果的简单合并是满足需求的。如果有这些函数, 那么查询结果就需要做进一步的处理而不是简单的合并, 如 sum 函数, 就需要对查询结果中的 sum 值再做一次累加返回给客户端, 这样客户端就能得到正确的结果而不需要应用再做二次处理。基于此原理, 代理需要就包含一些函数的 SQL 的执行结果做相应的逻辑处理, 然后返回给客户端, 以期实现分布式数据库代理的结果集合并处理。

[0003] 基于现有系统处理的缺点:

[0004] SQL 中有 sum 函数时, 返回的结果是单个节点 sum 值的简单合并。

[0005] SQL 中有 count 函数时, 返回的结果是单个节点 count 值的简单合并。

[0006] SQL 中有 max 函数时, 返回的结果是单个节点 max 值的简单合并。

[0007] SQL 中有 min 函数时, 返回的结果是单个节点 min 值的简单合并。

[0008] SQL 中有 avg 函数时, 返回的结果是单个节点 avg 值的简单合并。

[0009] SQL 中有 order by 函数时, 返回的结果是单个节点 order by 值的简单合并。

[0010] SQL 中有 group by 函数时, 返回的结果是单个节点 group by 值的简单合并。

[0011] SQL 中有 distinct 函数时, 返回的结果是单个节点的 distinct 值的简单合并。

[0012] 在互联网、电信等许多关键领域, 随着企业业务的不断发展, 企业的 I/T 环境也在不断进行演变, 单个数据库已经很难满足海量数据库存储和高并发数据访问, 分布式数据库的出现成为必然, 分布式数据库能够有效的解决海量数据存储和并发问题, 通过数据的水平切分和数据表的垂直切分, 有效的缓解了海量数据存储, 通过负载均衡、高可用、轮询等缓解了高并发问题, 但是分布式数据库在对待有函数的 SQL 执行结果时只是进行简单的合并, 从而客户端得到的是有歧义的结果。

发明内容

[0013] 本发明所要解决的技术问题是提供一种针对不同函数, 采用不同处理和合并方法的分布式数据库的代理结果合并方法。

[0014] 本发明解决上述技术问题的技术方案如下:一种分布式数据库的代理结果合并方法, 具体包括以下步骤:

[0015] 步骤 1: 解析第一 SQL 语句得到其包括的函数;

[0016] 步骤 2: 判断所述函数属于第一类函数还是第二类函数, 如果是第一类函数, 执行步骤 8; 否则, 执行步骤 3;

[0017] 步骤 3 :判断所述函数是否是 distinct 函数,如果是,执行步骤 5 ;否则,执行步骤 4 ;

[0018] 步骤 4 :对第一 SQL 语句进行预处理得到第二 SQL 语句,将第二 SQL 语句替换第一 SQL 语句;

[0019] 步骤 5 :发送第一 SQL 语句或第二 SQL 语句到 MYSQL 服务端;

[0020] 步骤 6 :获得 MYSQL 服务端返回的结果集,根据 MYSQL 数据包的协议格式,将结果集转换成数据包集合;

[0021] 步骤 7 :根据所述函数对数据包集合进行操作,得到新的结果集;

[0022] 步骤 8 :对新的结果集进行数据包封装得到新的数据包,将新的数据包发送到客户端。

[0023] 本发明的有益效果是:本发明具有经济性:支持在不增加硬件投入的条件下,提高系统并发处理能力;充分利用设备闲置时段的计算资源,提高系统资源利用率;利用低端服务器上负载分担数据库的存储空间进行数据处理;本发明还能提升性能和安全性:可配置方式移动计算、移动数据进行云计算,提供并行计算的性能和效率;保障数据的一致性、完整性、可用性,提高数据的隔离型,为高并发系统奠定基础。

[0024] 在上述技术方案的基础上,本发明还可以做如下改进。

[0025] 进一步,所述第一类函数包括 sum 函数、count 函数、max 函数和 min 函数;所述第二类函数包括 avg 函数、order by 函数、group by 函数和 distinct 函数。

[0026] 进一步,所述步骤 4 具体包括以下步骤:

[0027] 步骤 4.1 :判断第一 SQL 语句是否包含 avg 函数,如果是,执行步骤 4.2 ;否则执行步骤 4.3 ;

[0028] 步骤 4.2 :将第一 SQL 语句中的 avg() 改变为 sum(), count() 和 avg(), 得到第二 SQL 语句,执行步骤 5 ;

[0029] 步骤 4.3 :判断第一 SQL 语句是否包含 order by 函数,如果是,执行步骤 4.4 ;否则,执行步骤 4.6 ;

[0030] 步骤 4.4 :判断查询字段是否包含 order by 排序字段;如果是,执行步骤 4.6 ;否则,执行步骤 4.5 ;

[0031] 步骤 4.5 :将第一 SQL 语句中的排序字段添加到查询字段中形成第二 SQL 语句,执行步骤 5 ;

[0032] 步骤 4.6 :判断第一 SQL 语句是否包含 group by 函数;如果是,执行步骤 4.5 ;否则,执行步骤 5 。

[0033] 进一步,所述步骤 7 具体包括以下步骤:

[0034] 步骤 7.1 :判断所述函数是否是 sum 函数或 count 函数;如果是,执行步骤 7.2 ;否则,执行步骤 7.3 ;

[0035] 步骤 7.2 :对多节点的结果集进行累加构成新的结果集,执行步骤 8 ;

[0036] 步骤 7.3 :判断所述函数是否是 max 函数;如果是,执行步骤 7.4 ;否则,执行步骤 7.5 ;

[0037] 步骤 7.4 :对多节点的结果集进行求最大值得到新的结果集,执行步骤 8 ;

[0038] 步骤 7.5 :判断所述函数是否是 min 函数;如果是,执行步骤 7.6 ;否则执行步骤

7.7；

- [0039] 步骤 7.6：对多节点的结果集进行求最小值得到新的结果集，执行步骤 8；
- [0040] 步骤 7.7：判断所述函数是否是 avg 函数；如果是，执行步骤 7.8；否则，执行步骤 7.9；
- [0041] 步骤 7.8：对所有节点的结果集进行统一求平均值得到新的结果集，执行步骤 8；
- [0042] 步骤 7.9：判断所述函数是否是 order by 函数，如果是，执行步骤 7.10；否则，执行步骤 7.11；
- [0043] 步骤 7.10：对多节点的结果集进行重新排序得到新的结果集，执行步骤 8；
- [0044] 步骤 7.11：判断所述函数是否是 group by 函数，如果是，执行步骤 7.12；否则，执行步骤 7.13；
- [0045] 步骤 7.12：对多节点的结果集进行重新组装得到新的结果集，执行步骤 8；
- [0046] 步骤 7.13：对单个节点的 distinct 值进行简单合并，得到新的结果集，执行步骤 8。
- [0047] 进一步，所述步骤 7.2 包括以下步骤：
 - [0048] 步骤 7.2.1：从结果集中获得列字段的列数据包，在列数据包中找到 sum() 字段或 count() 字段，并获得该字段的序列号；
 - [0049] 步骤 7.2.2：从结果集中获得行字段的行数据包，在行数据包中，获得每一行的 sum() 函数或 count() 函数的结果，并进行累加得到累加值；
 - [0050] 步骤 7.2.3：将累加值放到第一行数据包里，删除多余行数据包构成新的结果集，并将新的结果集里代表包序列号的信息更改为正确的序列号的信息，执行步骤 8。
- [0051] 进一步，所述所述步骤 7.4 具体包括以下步骤：
 - [0052] 步骤 7.4.1：从结果集中获得列字段的列数据包，在列数据包中找到 max() 字段，并获得该字段的序列号；
 - [0053] 步骤 7.4.2：从结果集中获得行字段的行数据包，在行数据包中查找每一行的 max() 函数的结果，并比较大小获得其最大值；
 - [0054] 步骤 7.4.3：将最大值放到第一行数据包里，并删除多余行数据包构成新的结果集，并将新的结果集里代表包序列号的信息更改为正确的序列号的信息，执行步骤 8。
- [0055] 进一步，所述步骤 7.6 具体包括以下步骤：
 - [0056] 步骤 7.6.1：从结果集中获得列字段的列数据包，在列数据包中找到 min() 字段，并获得该字段的序列号；
 - [0057] 步骤 7.6.2：从结果集中获得行字段的行数据包，在行数据包中查找每一行的 min() 函数的结果，并比较大小获得其最小值；
 - [0058] 步骤 7.6.3：将最小值放到第一行数据包里，并删除多余行数据包构成新的结果集，并将新的结果集里代表包序列号的信息更改为正确的序列号的信息，执行步骤 8。
- [0059] 进一步，所述步骤 7.8 具体包括以下步骤：
 - [0060] 步骤 7.8.1：从结果集中获得列字段的列数据包，在列数据包中找到 avg()、sum() 和 count() 字段，并获得该字段的序列号；
 - [0061] 步骤 7.8.2：从结果集中获得行字段的行数据包，在行数据包中获得每一行的 sum 函数和 count 函数的结果，并且分别累加得到总的 sum 值和 count 值；

- [0062] 步骤 7.8.3 :sum 值除以 count 值得到平均值,并组装成行数据包;
- [0063] 步骤 7.8.4 :删除多余行数据包和多余列数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8。
- [0064] 进一步,所述步骤 7.10 具体包括以下步骤:
- [0065] 步骤 7.10.1 :从结果集中获得列字段的列数据包,查找得到包含排序字段的列字段的序列号;
- [0066] 步骤 7.10.2 :从结果集中获得行数据包,从行数据包中获得每一行的排序字段的值,并且将排序字段的值保持到结果集中;
- [0067] 步骤 7.10.3 :判断所述第二 SQL 语句中是否包含 desc,如果是,执行步骤 7.10.4 ;否则,执行步骤 7.10.5 ;
- [0068] 步骤 7.10.4 :对结果集中的值进行降序排列得到新的结果集,在新的结构集中代表包序列号的信息更改为正确的序列号的信息,执行步骤 8;
- [0069] 步骤 7.10.5 :对结果集中的值进行升序排列并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8。
- [0070] 进一步,所述步骤 7.12 具体包括以下步骤:
- [0071] 步骤 7.12.1 :从结果集中获得列字段的列数据包,在列数据包中查找得到包含 group by 字段的列字段的序列号;
- [0072] 步骤 7.12.2 :从结果集中获得行字段的行数据包,并获得每一行的排序字段的值,将排序字段的值存入结果集中;
- [0073] 步骤 7.12.3 :对结果集中的值进行升序排列,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8。
- [0074] 进一步,所述步骤 7.13 具体包括以下步骤:
- [0075] 步骤 7.13.1 :从结果集中获得列字段的列数据包,在列数据包中查找得到包含 distinct 字段的列字段的序列号;
- [0076] 步骤 7.13.2 :从结果集中获得行字段的行数据包,并获得每一行的排序字段的值,将排序字段的值存入结果集中;
- [0077] 步骤 7.13.3 :对结果集中的值进行筛选剔除重复的数据,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息,执行步骤 8。
- [0078] 本发明所要解决的技术问题是提供一种针对不同函数,采用不同处理和合并方法的分布式数据库的代理结果合并系统。
- [0079] 本发明解决上述技术问题的技术方案如下:一种分布式数据库的代理结果合并系统,包括:解析 SQL 模块、解析数据包模块、逻辑处理模块、数据包组装模块和反馈模块;
- [0080] 所述解析 SQL 模块用于解析 SQL 语句得到其包括的函数;
- [0081] 所述解析数据包模块用于分析数据包,提取行数据,列字段数据等信息;
- [0082] 所述逻辑处理模块用于提取每行的对应函数所在字段的数据信息,并进行逻辑处理,根据不同函数进行对应的处理得到结果;如:对应值相加、求最大值、求最小值、求平均值或排序等;
- [0083] 所述数据包组装模块用于将处理后的结果重新封装成符合 MYSQL 协议的数据包,

并将所述数据包发送到反馈模块；

[0084] 所述反馈模块用于将数据包发送到客户端。

附图说明

[0085] 图 1 为本发明所述的一种分布式数据库的代理结果合并系统结构框图；

[0086] 图 2 为本发明所述的一种分布式数据库的代理结果合并方法流程图；

[0087] 图 3 为本发明所述 sum 函数处理流程图；

[0088] 图 4 为本发明所述 count 函数处理流程图；

[0089] 图 5 为本发明所述 max 函数处理流程图；

[0090] 图 6 为本发明所述 min 函数处理流程图；

[0091] 图 7 为本发明所述 avg 函数处理流程图；

[0092] 图 8 为本发明所述 order by 函数处理流程图；

[0093] 图 9 为本发明所述 group by 函数处理流程图；

[0094] 图 10 为本发明所述 distinct 函数处理流程图。

[0095] 附图中，各标号所代表的部件列表如下：

[0096] 1、解析 SQL 模块，2、解析数据包模块，3、逻辑处理模块，4、数据包组装模块，5、反馈模块。

具体实施方式

[0097] 以下结合附图对本发明的原理和特征进行描述，所举实例只用于解释本发明，并非用于限定本发明的范围。

[0098] 如图 1 所示，为本发明所述的一种分布式数据库的代理结果合并系统，包括：解析 SQL 模块 1、解析数据包模块 2、逻辑处理模块 3、数据包组装模块 4 和反馈模块 5；

[0099] 所述解析 SQL 模块 1 用于解析语句得到其包括的函数；

[0100] 所述解析数据包模块 2 用于分析数据包，提取行数据，列字段数据等信息；

[0101] 所述逻辑处理模块 3 用于提取每行的对应函数所在字段的数据信息，并进行逻辑处理，根据不同函数进行对应的处理得到结果；如：对应值相加、求最大值、求最小值、求平均值或排序等；

[0102] 所述数据包组装模块 4 用于将处理后的结果重新封装成符合 MYSQL 协议的数据包，并将所述数据包发送到反馈模块 5；

[0103] 所述反馈模块 5 用于将数据包发送到客户端。

[0104] 如图 2 所示，为本发明所述的一种分布式数据库的代理结果合并方法，具体包括以下步骤：

[0105] 步骤 1：解析第一 SQL 语句得到其包括的函数；

[0106] 步骤 2：判断所述函数属于第一类函数还是第二类函数，如果是第一类函数，执行步骤 8；否则，执行步骤 3；

[0107] 步骤 3：判断所述函数是否是 distinct 函数，如果是，执行步骤 5；否则，执行步骤 4；

[0108] 步骤 4：对第一 SQL 语句进行预处理得到第二 SQL 语句，将第二 SQL 语句替换第一

SQL 语句；

[0109] 步骤 5：发送第一 SQL 语句或第二 SQL 语句到 MYSQL 服务端；

[0110] 步骤 6：获得 MYSQL 服务端返回的结果集，根据 MYSQL 数据包的协议格式，将结果集转换成数据包集合；

[0111] 步骤 7：根据所述函数对数据包集合进行操作，得到新的结果集；

[0112] 步骤 8：对新的结果集进行数据包封装得到新的数据包，将新的数据包发送到客户端。

[0113] 所述第一类函数包括 sum 函数、count 函数、max 函数和 min 函数；所述第二类函数包括 avg 函数、order by 函数、group by 函数和 distinct 函数。

[0114] 如图 3 所示，为本发明所述 sum 函数处理流程图；

[0115] 步骤 3.1：获得 MYSQL 服务端返回的结果集，根据 MYSQL 数据包的协议格式，将结果集转换成数据包集合，数据包集合中包括反馈结果的列字段包和行数据包，便于后续结果集的合并整理；

[0116] 步骤 3.2：判断所述函数是否是 sum 函数；如果是，执行步骤 3.3；否则，执行步骤 3.6；

[0117] 步骤 3.3：从结果集中获得列字段的列数据包，在列数据包中找到 sum() 字段，并获得该字段的序列号；

[0118] 步骤 3.4：从结果集中获得行字段的行数据包，在行数据包中，获得每一行的 sum() 函数的结果，并进行累加得到累加值；

[0119] 步骤 3.5：将累加值放到第一行数据包里，删除多余行数据包构成新的结果集，并将新的结果集里代表包序列号的信息更改为正确的序列号的信息；

[0120] 步骤 3.6：对新的结果集进行数据包封装得到新的数据包，将新的数据包发送到客户端。

[0121] 如图 4 所示，为本发明所述 count 函数处理流程图；

[0122] 步骤 4.1：获得 MYSQL 服务端返回的结果集，根据 MYSQL 数据包的协议格式，将结果集转换成数据包集合，数据包集合中包括反馈结果的列字段包和行数据包，便于后续结果集的合并整理；

[0123] 步骤 4.2：判断所述函数是否是 count 函数；如果是，执行步骤 4.3；否则，执行步骤 4.6；

[0124] 步骤 4.3：从结果集中获得列字段的列数据包，在列数据包中找到 count() 字段，并获得该字段的序列号；

[0125] 步骤 4.4：从结果集中获得行字段的行数据包，在行数据包中，获得每一行的 count() 函数的结果，并进行累加得到累加值；

[0126] 步骤 4.5：将累加值放到第一行数据包里，删除多余行数据包构成新的结果集，并将新的结果集里代表包序列号的信息更改为正确的序列号的信息；

[0127] 步骤 4.6：对新的结果集进行数据包封装得到新的数据包，将新的数据包发送到客户端。

[0128] 如图 5 所示，为本发明所述 max 函数处理流程图；

[0129] 步骤 5.1：获得 MYSQL 服务端返回的结果集，根据 MYSQL 数据包的协议格式，将结

果集转换成数据包集合,数据包集合中包括反馈结果的列字段包和行数据包,便于后续结果集的合并整理;

[0130] 步骤 5.2 :判断所述函数是否是 max 函数;如果是,执行步骤 5.3 ;否则,执行步骤 5.6 ;

[0131] 步骤 5.3 :从结果集中获得列字段的列数据包,在列数据包中找到 max() 字段,并获得该字段的序列号;

[0132] 步骤 5.4 :从结果集中获得行字段的行数据包,在行数据包中查找每一行的 max() 函数的结果,并比较大小获得其最大值;

[0133] 步骤 5.5 :将最大值放到第一行数据包里,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息;

[0134] 步骤 5.6 :对新的结果集进行数据包封装得到新的数据包,将新的数据包发送到客户端。

[0135] 如图 6 所示,为本发明所述 min 函数处理流程图;

[0136] 步骤 6.1 :获得 MYSQL 服务端返回的结果集,根据 MYSQL 数据包的协议格式,将结果集转换成数据包集合,数据包集合中包括反馈结果的列字段包和行数据包,便于后续结果集的合并整理;

[0137] 步骤 6.2 :判断所述函数是否是 min 函数;如果是,执行步骤 6.3 ;否则,执行步骤 6.6 ;

[0138] 步骤 6.3 :从结果集中获得列字段的列数据包,在列数据包中找到 min() 字段,并获得该字段的序列号;

[0139] 步骤 6.4 :从结果集中获得行字段的行数据包,在行数据包中查找每一行的 min() 函数的结果,并比较大小获得其最小值;

[0140] 步骤 6.5 :将最小值放到第一行数据包里,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息;

[0141] 步骤 6.6 :对新的结果集进行数据包封装得到新的数据包,将新的数据包发送到客户端。

[0142] 如图 7 所示,为本发明所述 avg 函数处理流程图;

[0143] 步骤 7.1 :判断第一 SQL 语句是否包含 avg 函数,如果是,执行步骤 7.2 ;否则执行步骤 7.3 ;

[0144] 步骤 7.2 :将第一 SQL 语句中的 avg() 改变为 sum(), count() 和 avg(), 得到第二 SQL 语句;

[0145] 步骤 7.3 :发送第二 SQL 语句到 MYSQL 服务端;

[0146] 步骤 7.4 :获得 MYSQL 服务端返回的结果集,根据 MYSQL 数据包的协议格式,将结果集转换成数据包集合,数据包集合中包括反馈结果的列字段包和行数据包,便于后续结果集的合并整理;

[0147] 步骤 7.5 :从结果集中获得列字段的列数据包,在列数据包中找到 avg()、sum() 和 count() 字段,并获得该字段的序列号;

[0148] 步骤 7.6 :从结果集中获得行字段的行数据包,在行数据包中获得每一行的 sum 函数和 count 函数的结果,并且分别累加得到总的 sum 值和 count 值;

- [0149] 步骤 7.7 :sum 值除以 count 值得到平均值,并组装成行数据包 ;
- [0150] 步骤 7.8 :删除多余行数据包和多余列数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息 ;
- [0151] 步骤 7.9 :对新的结果集进行数据包封装得到新的数据包,将新的数据包发送到客户端。
- [0152] 如图 8 所示,为本发明所述 order by 函数处理流程图 ;
- [0153] 步骤 8.1 :判断第一 SQL 语句是否包含 order by 函数,如果是,执行步骤 8.2 ;否则,执行步骤 8.4 ;
- [0154] 步骤 8.2 :判断查询字段是否包含 order by 排序字段 ;如果是,执行步骤 8.3 ;否则,执行步骤 8.4 ;
- [0155] 步骤 8.3 :将第一 SQL 语句中的排序字段添加到查询字段中形成第二 SQL 语句 ;
- [0156] 步骤 8.4 :发送第二 SQL 语句到 MYSQL 服务端 ;
- [0157] 步骤 8.5 :获得 MYSQL 服务端返回的结果集,根据 MYSQL 数据包的协议格式,将结果集转换成数据包集合,数据包集合中包括反馈结果的列字段包和行数据包,便于后续结果集的合并整理 ;
- [0158] 步骤 8.6 :从结果集中获得列字段的列数据包,查找得到包含排序字段的列字段的序列号 ;
- [0159] 步骤 8.7 :从结果集中获得行数据包,从行数据包中获得每一行的排序字段的值,并且将排序字段的值保持到结果集中 ;
- [0160] 步骤 8.8 :判断所述第二 SQL 语句中是否包含 desc,如果是,执行步骤 8.9 ;否则,执行步骤 8.10 ;
- [0161] 步骤 8.9 :对结果集中的值进行降序排列得到新的结果集,在新的结构集中代表包序列号的信息更改为正确的序列号的信息,执行步骤 8.11 ;
- [0162] 步骤 8.10 :对结果集中的值进行升序排列并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息 ;
- [0163] 步骤 8.11 :对新的结果集进行数据包封装得到新的数据包,将新的数据包发送到客户端。
- [0164] 如图 9 所示,为本发明所述 group by 函数处理流程图 ;
- [0165] 步骤 9.1 :判断第一 SQL 语句是否包含 group by 函数 ;如果是,执行步骤 9.2 ;否则,执行步骤 9.3 ;
- [0166] 步骤 9.2 :将第一 SQL 语句中的排序字段添加到查询字段中形成第二 SQL 语句 ;
- [0167] 步骤 9.3 :发送第二 SQL 语句到 MYSQL 服务端 ;
- [0168] 步骤 9.4 :获得 MYSQL 服务端返回的结果集,根据 MYSQL 数据包的协议格式,将结果集转换成数据包集合,数据包集合中包括反馈结果的列字段包和行数据包,便于后续结果集的合并整理 ;
- [0169] 步骤 9.5 :从结果集中获得列字段的列数据包,在列数据包中查找得到包含 group by 字段的列字段的序列号 ;
- [0170] 步骤 9.6 :从结果集中获得行字段的行数据包,并获得每一行的排序字段的值,将排序字段的值存入结果集中 ;

[0171] 步骤 9.7 :对结果集中的值进行升序排列,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息;

[0172] 步骤 9.8 :对新的结果集进行数据包封装得到新的数据包,将新的数据包发送到客户端。

[0173] 如图 10 所示,为本发明所述 distinct 函数处理流程图;

[0174] 步骤 10.1 :发送第一 SQL 语句到 MYSQL 服务端;

[0175] 步骤 10.2 :获得 MYSQL 服务端返回的结果集,根据 MYSQL 数据包的协议格式,将结果集转换成数据包集合,数据包集合中包括反馈结果的列字段包和行数据包,便于后续结果集的合并整理;

[0176] 步骤 10.3 :从结果集中获得列字段的列数据包,在列数据包中查找得到包含 distinct 字段的列字段的序列号;

[0177] 步骤 10.4 :从结果集中获得行字段的行数据包,并获得每一行的排序字段的值,将排序字段的值存入结果集中;

[0178] 步骤 10.5 :对结果集中的值进行筛选剔除重复的数据,并删除多余行数据包构成新的结果集,并将新的结果集里代表包序列号的信息更改为正确的序列号的信息;

[0179] 步骤 10.6 :对新的结果集进行数据包封装得到新的数据包,将新的数据包发送到客户端。

[0180] 以上所述仅为本发明的较佳实施例,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

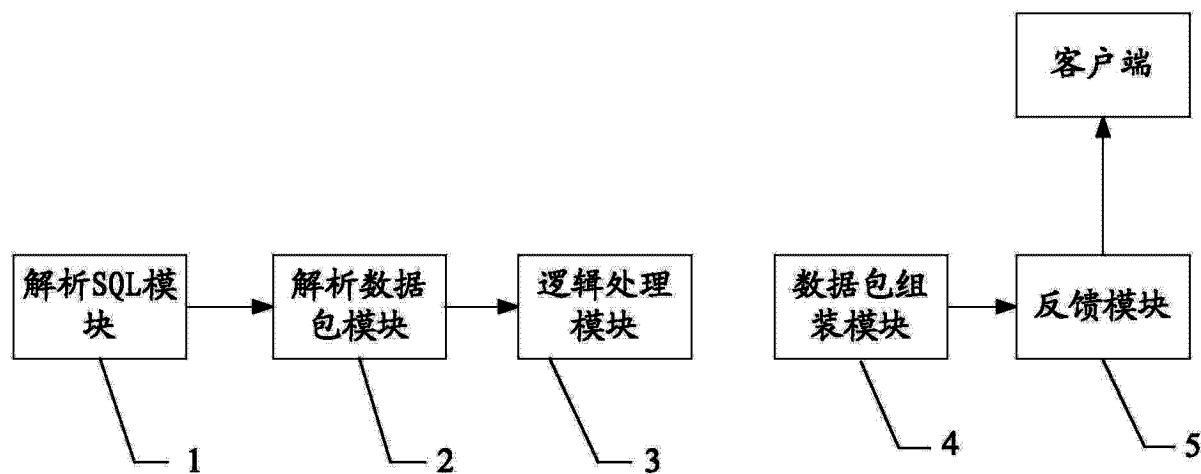


图 1

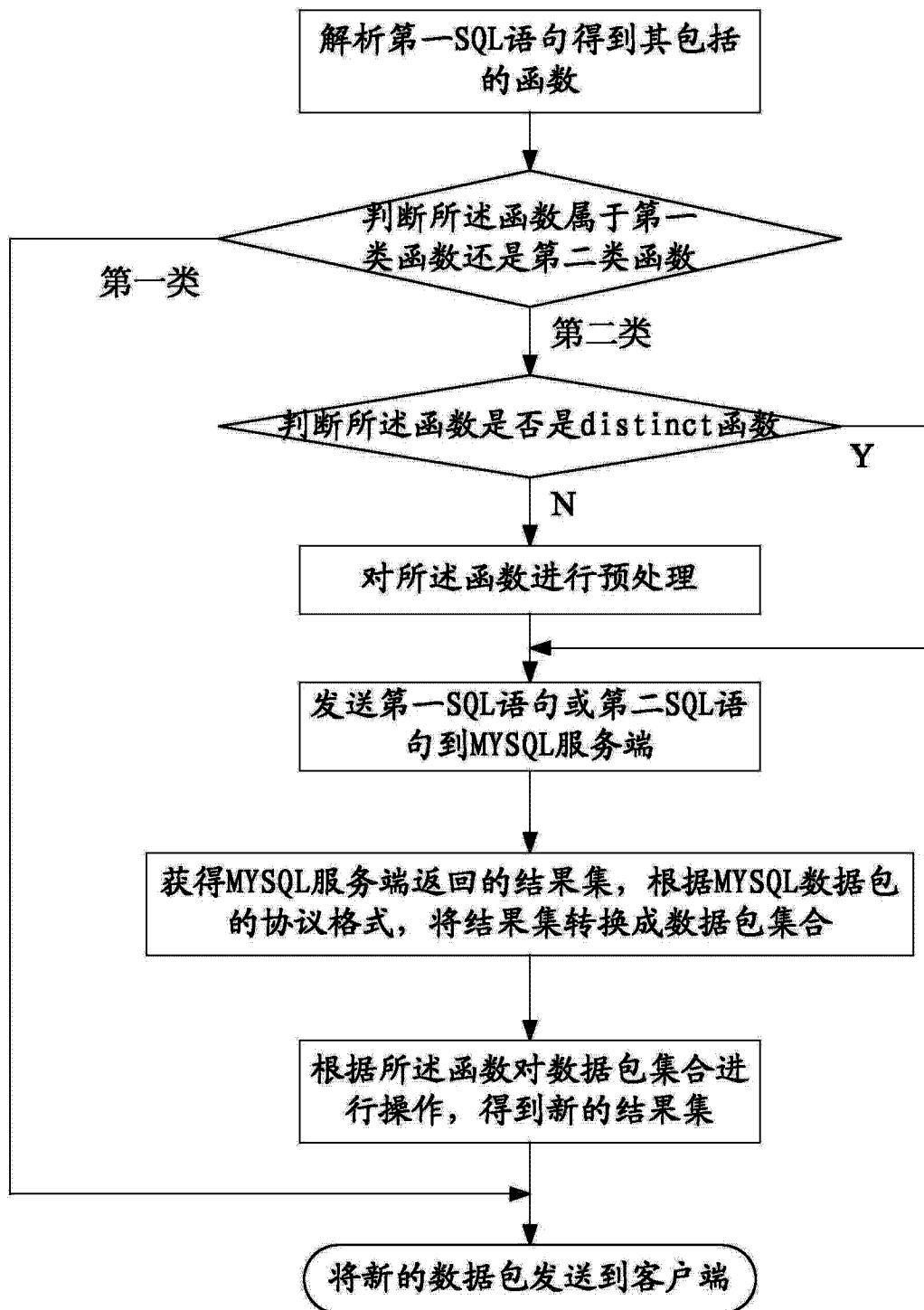


图 2

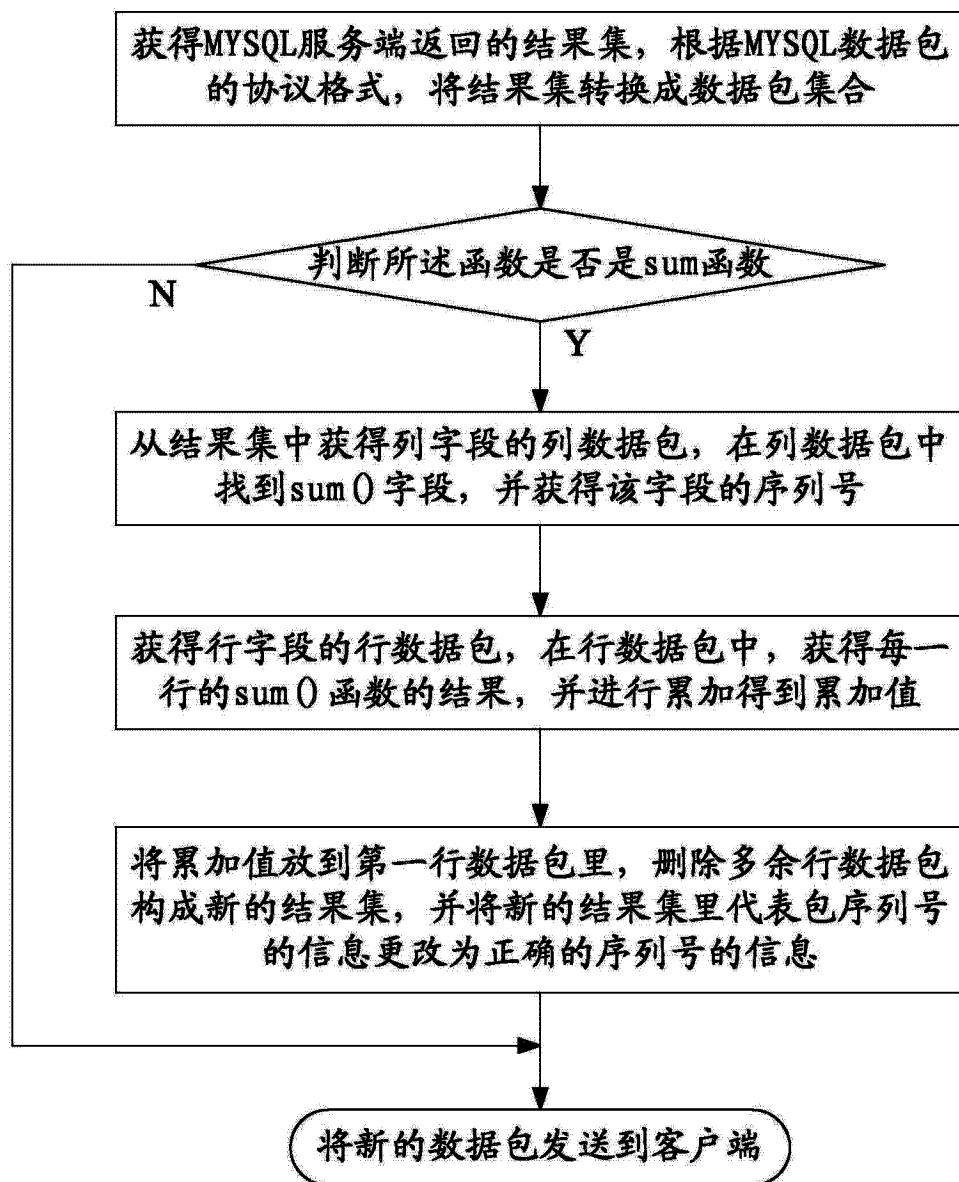


图 3

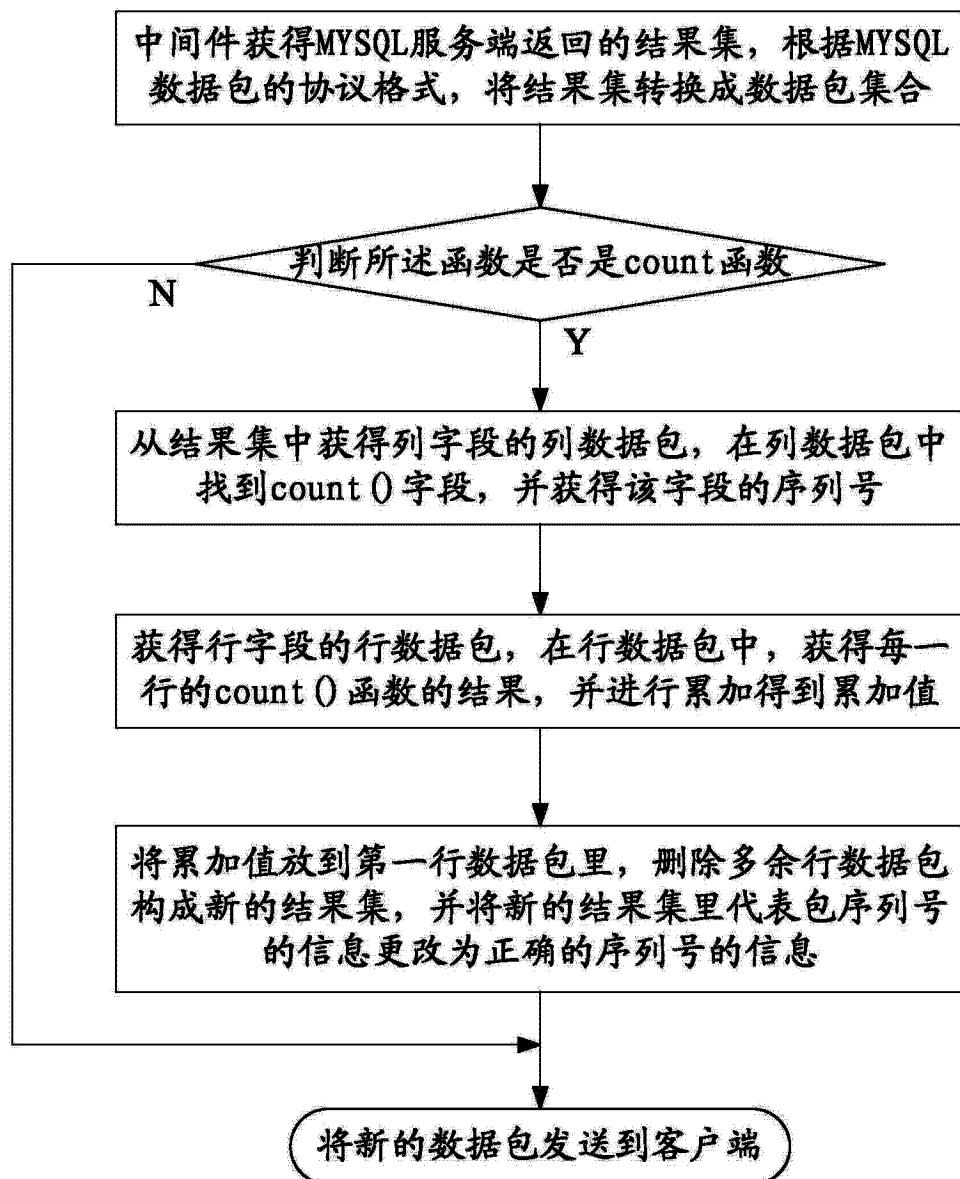


图 4

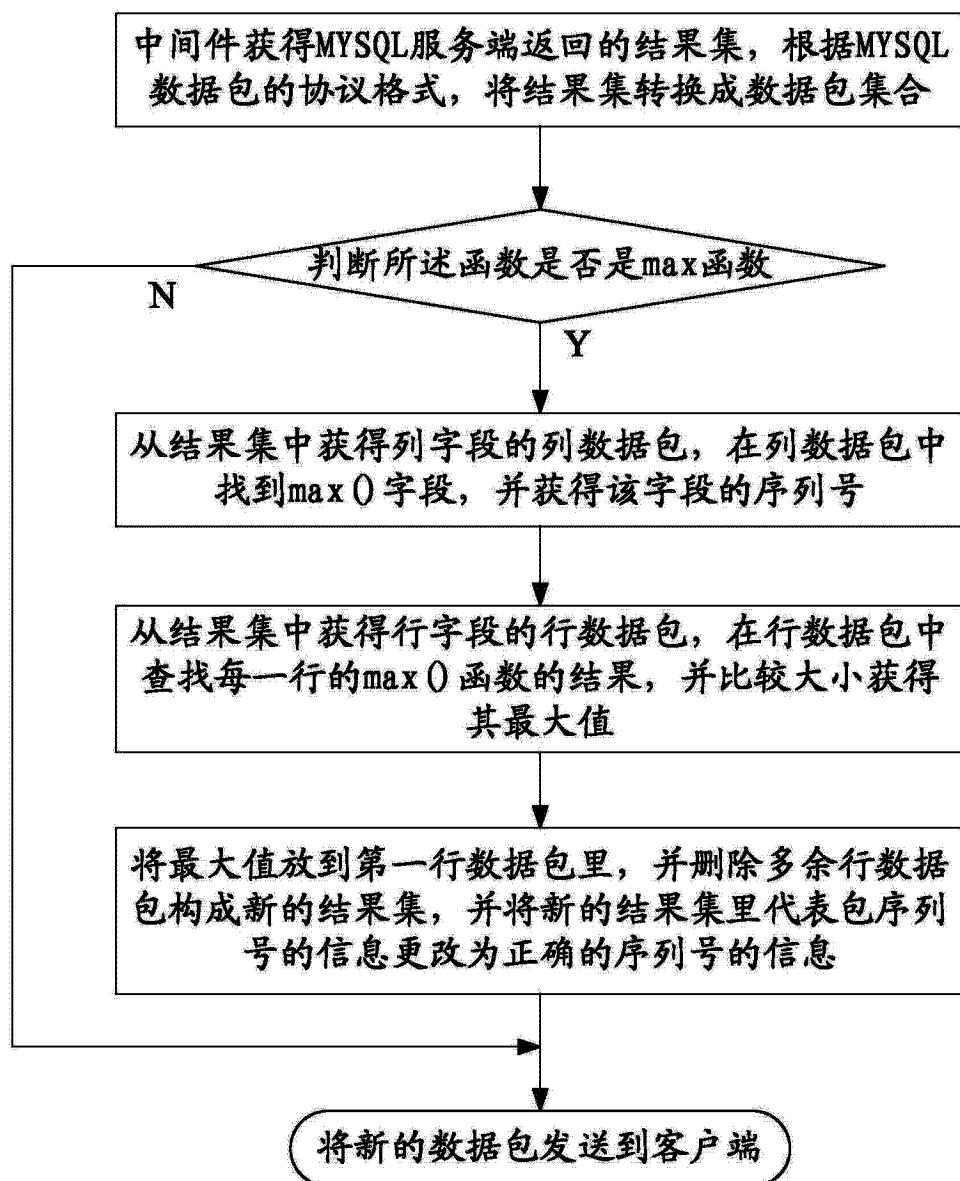


图 5

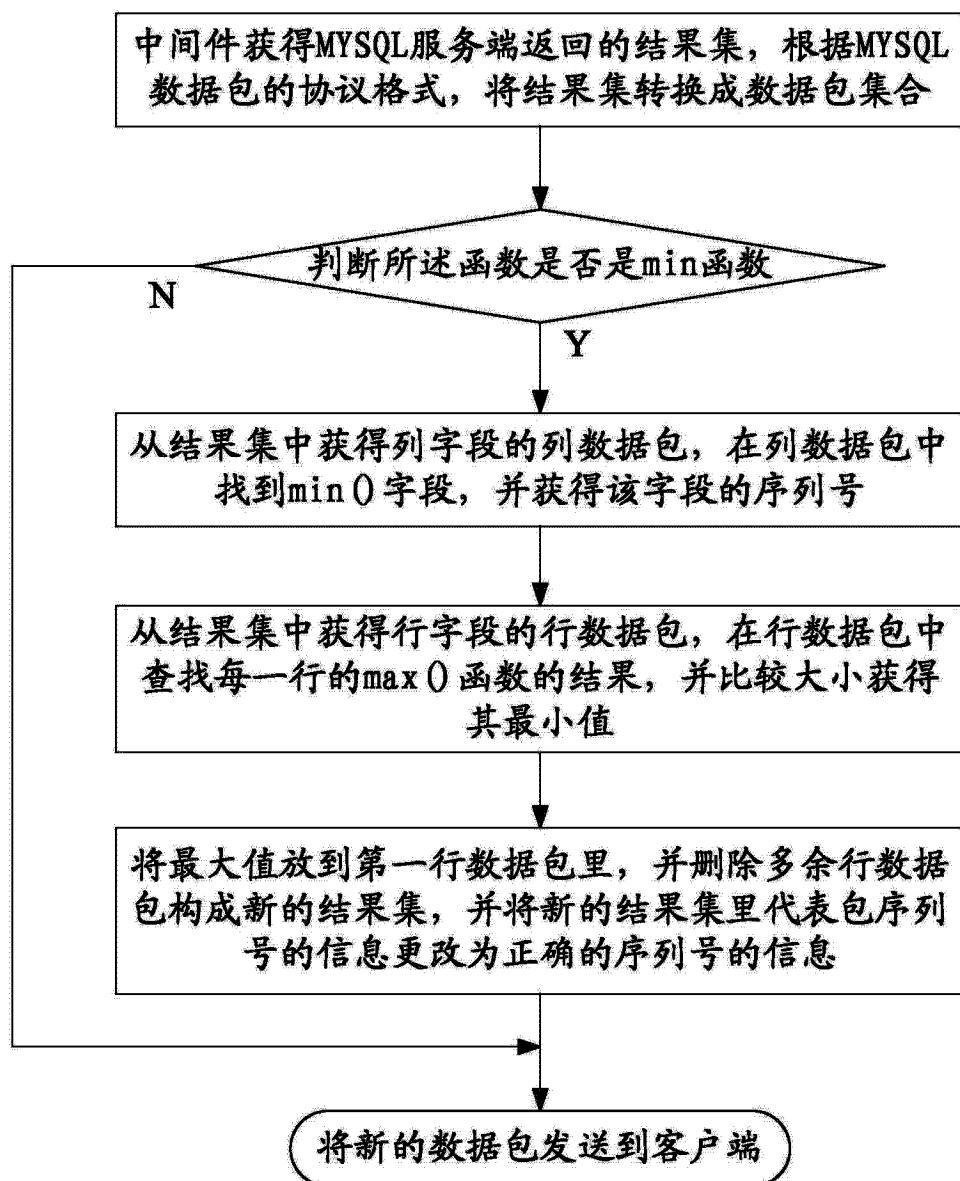


图 6

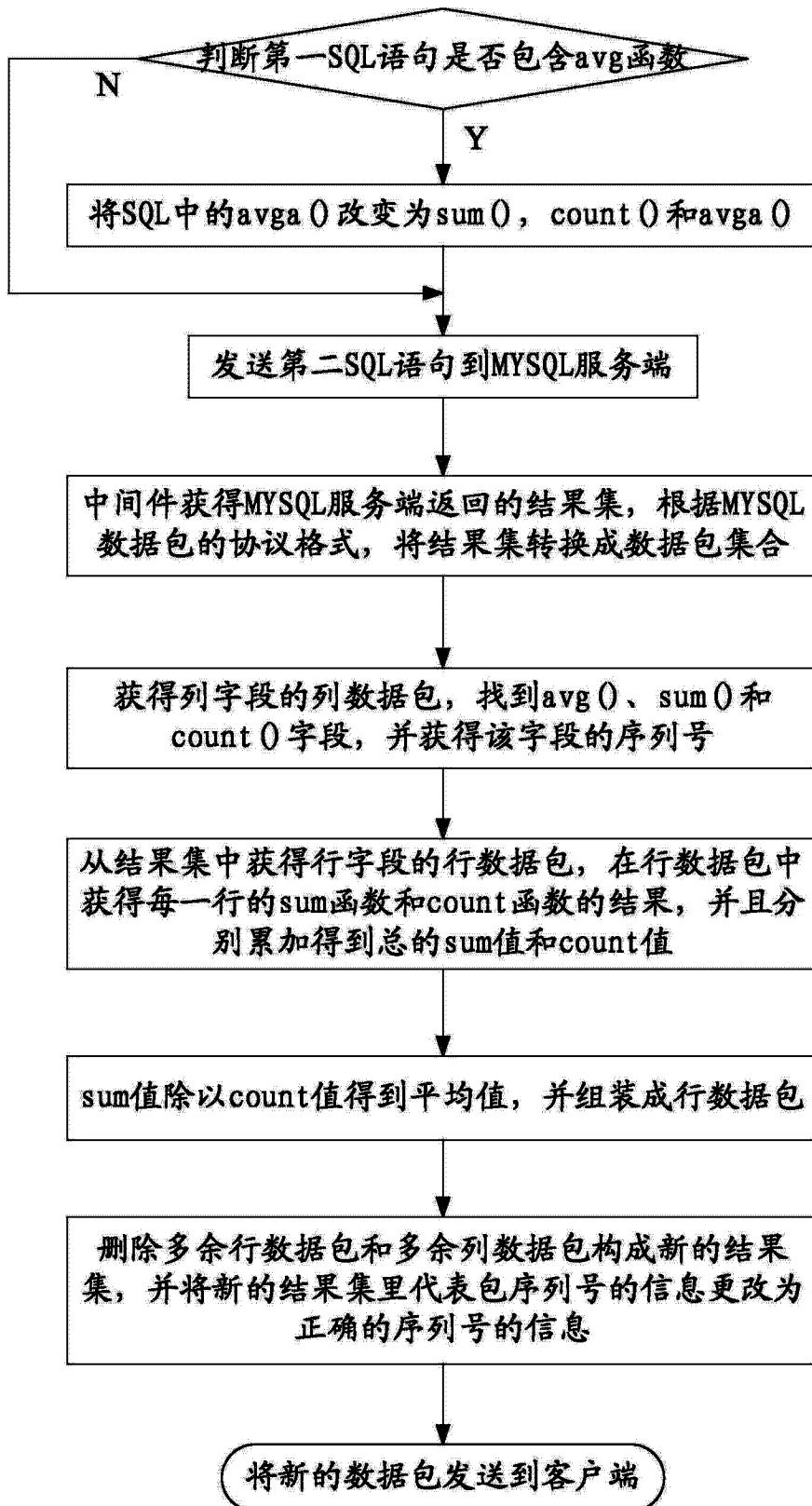


图 7

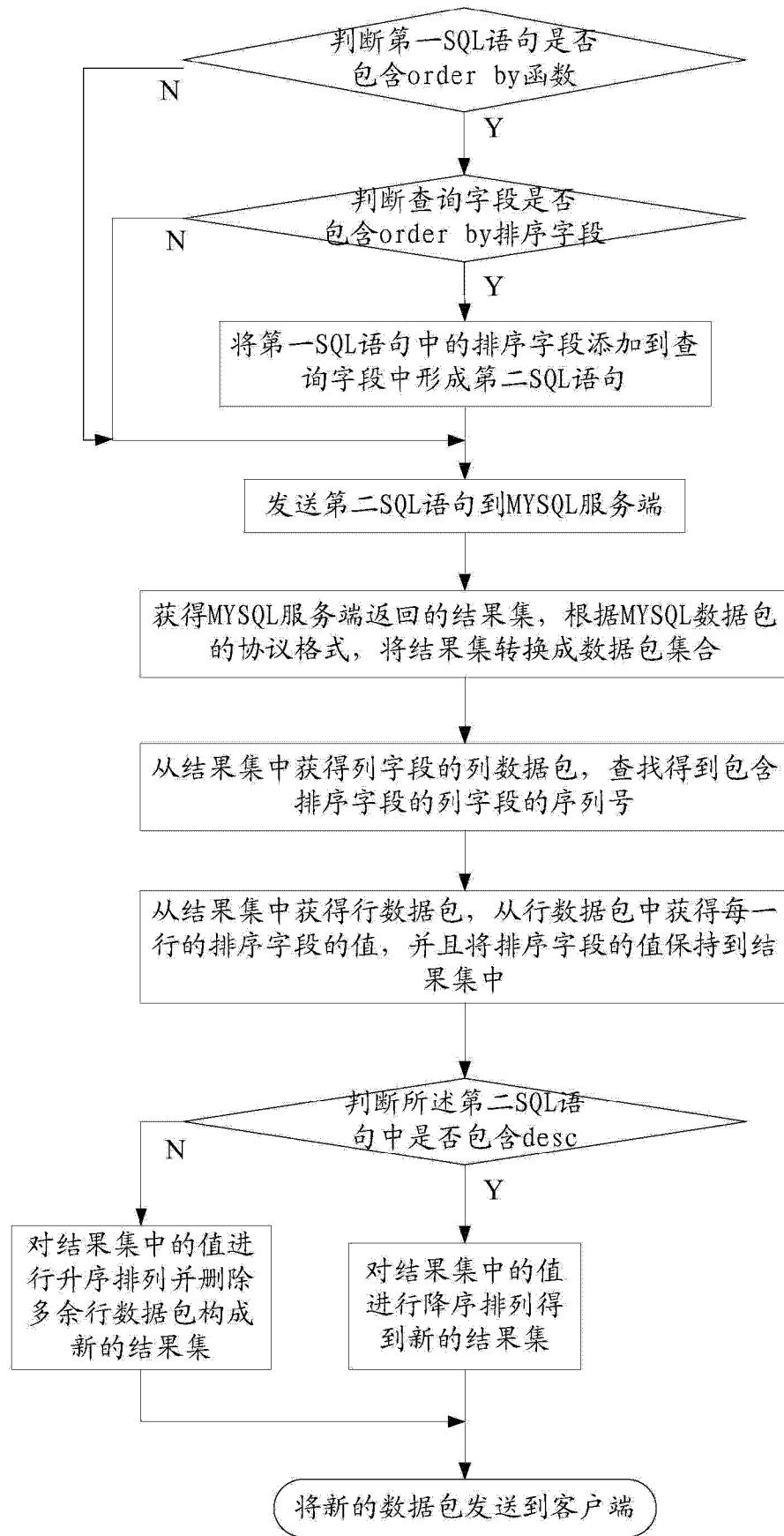


图 8

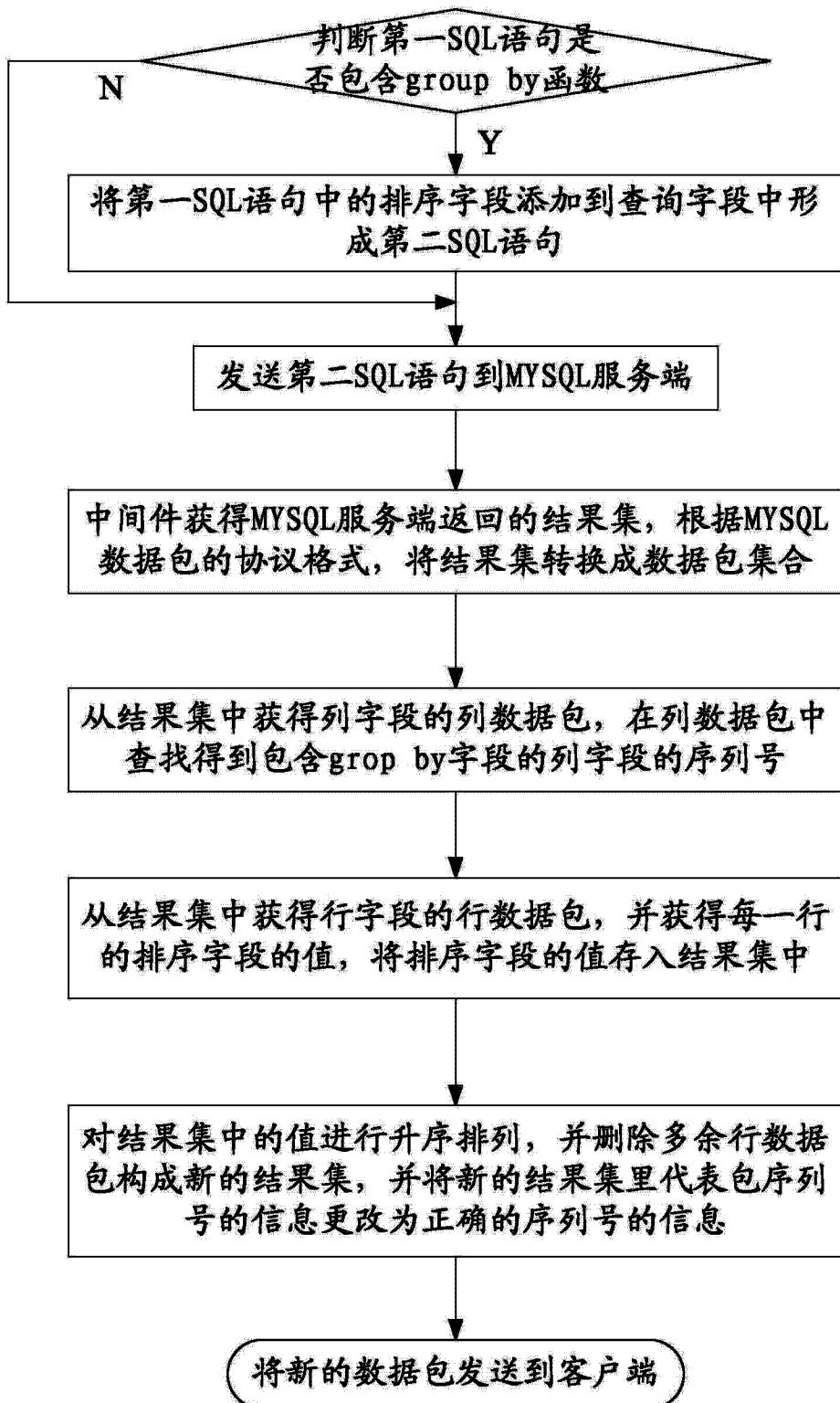


图 9

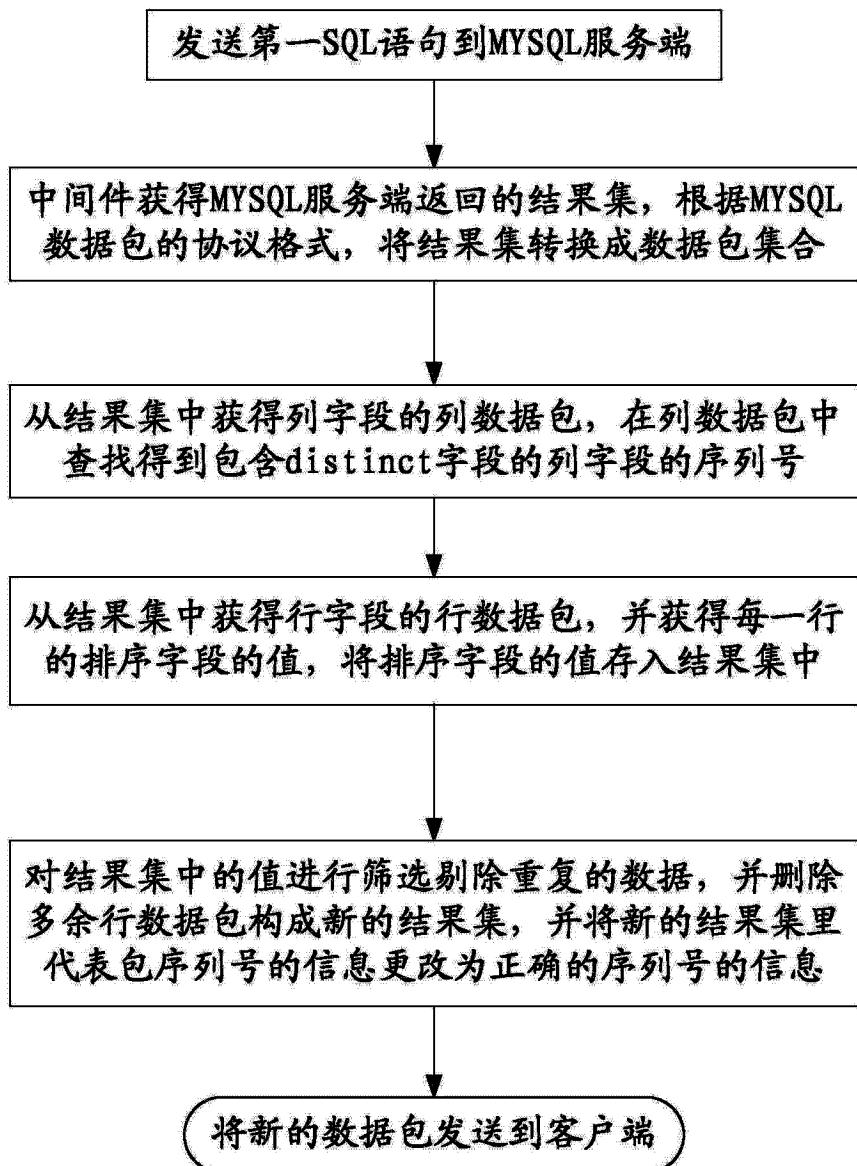


图 10