

(19) 中华人民共和国国家知识产权局



(12) 发明专利申请

(10) 申请公布号 CN 104285222 A

(43) 申请公布日 2015.01.14

(21) 申请号 201380023462.9

(51) Int. Cl.

(22) 申请日 2013.05.02

G06F 17/30 (2006.01)

(30) 优先权数据

13/465,769 2012.05.07 US

(85) PCT国际申请进入国家阶段日

2014.11.04

(86) PCT国际申请的申请数据

PCT/IB2013/053489 2013.05.02

(87) PCT国际申请的公布数据

W02013/168058 EN 2013.11.14

(71) 申请人 国际商业机器公司

地址 美国纽约

(72) 发明人 S·辛格 R·苏玛

(74) 专利代理机构 北京市中咨律师事务所

11247

代理人 于静 张亚非

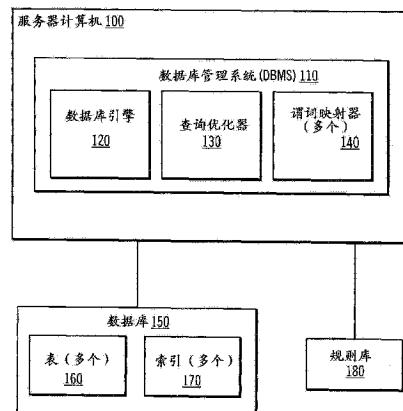
权利要求书2页 说明书11页 附图11页

(54) 发明名称

使用谓词映射器优化查询

(57) 摘要

提供一种用于重写查询的计算机实现的方法、计算机程序产品和系统。接收包括查询谓词的查询。确定所述查询谓词包括具有关联的谓词映射器的一个列。使用提供从所述一个列到另一个列的映射的所述谓词映射器，针对所述查询谓词生成等效谓词。评估具有所述等效谓词的所述查询的成本。响应于确定所述查询的所述成本是最优的，重写所述查询以便使用所述等效谓词替换所述查询谓词。



1. 一种用于重写查询的方法,包括 :

通过计算机的处理器接收包括查询谓词的查询 ;

确定所述查询谓词包括具有关联的谓词映射器的一个列 ;

使用提供从所述一个列到另一个列的映射的所述谓词映射器,针对所述查询谓词生成等效谓词 ;

评估具有所述等效谓词的所述查询的成本 ;以及

响应于确定所述查询的所述成本是最优的,重写所述查询以便使用所述等效谓词替换所述查询谓词。

2. 根据权利要求 1 的方法,其中所述列包括生成列。

3. 根据权利要求 2 的方法,还包括 :

在规则库中查找定义所述生成列的函数的反函数 ;以及

使用所述反函数创建所述谓词映射器。

4. 根据权利要求 2 的方法,还包括 :

确定定义所述生成列的函数可逆 ;

使用数学方程式查找所述函数的反函数 ;以及

使用所述反函数创建所述谓词映射器。

5. 根据权利要求 4 的方法,其中确定所述函数可逆进一步包括 :

确定所述函数是一对一的。

6. 根据权利要求 1 的方法,其中所述谓词映射器包括模式对象,并且定义两个或更多列之间的关系。

7. 根据权利要求 1 的方法,其中由用户创建所述谓词映射器。

8. 根据权利要求 1 的方法,其中由数据库引擎创建所述谓词映射器。

9. 一种用于重写查询的计算机程序产品,所述计算机程序产品包括 :

计算机可读存储介质,其中包含计算机可读程序代码,其中当由计算机的处理器执行时,所述计算机可读程序代码被配置为执行以下操作 :

接收包括查询谓词的查询 ;

确定所述查询谓词包括具有关联的谓词映射器的一个列 ;

使用提供从所述一个列到另一个列的映射的所述谓词映射器,针对所述查询谓词生成等效谓词 ;

评估具有所述等效谓词的所述查询的成本 ;以及

响应于确定所述查询的所述成本是最优的,重写所述查询以便使用所述等效谓词替换所述查询谓词。

10. 根据权利要求 9 的计算机程序产品,其中所述一个列包括生成列。

11. 根据权利要求 10 的计算机程序产品,其中当由所述计算机的所述处理器执行时,所述计算机可读程序代码被配置为执行以下操作 :

在规则库中查找定义所述生成列的函数的反函数 ;以及

使用所述反函数创建所述谓词映射器。

12. 根据权利要求 10 的计算机程序产品,其中当由所述计算机的所述处理器执行时,所述计算机可读程序代码被配置为执行以下操作 :

确定定义所述生成列的函数可逆；

使用数学方程式查找所述函数的反函数；以及

使用所述反函数创建所述谓词映射器。

13. 根据权利要求 12 的计算机程序产品，其中当由所述计算机的所述处理器执行时，用于确定所述函数可逆的所述计算机可读程序代码被配置为执行以下操作：

确定所述函数是一对一的。

14. 根据权利要求 9 的计算机程序产品，其中所述谓词映射器包括模式对象，并且定义两个或更多列之间的关系。

15. 根据权利要求 9 的计算机程序产品，其中由用户创建所述谓词映射器。

16. 根据权利要求 9 的计算机程序产品，其中由数据库引擎创建所述谓词映射器。

17. 一种用于重写查询的计算机系统，包括：

处理器；以及

耦合到所述处理器的存储设备，其中所述存储设备具有存储在其上的程序，并且其中所述处理器被配置为执行所述程序的指令以便执行操作，其中所述操作包括：

接收包括查询谓词的查询；

确定所述查询谓词包括具有关联的谓词映射器的一个列；

使用提供从所述一个列到另一个列的映射的所述谓词映射器，针对所述查询谓词生成等效谓词；

评估具有所述等效谓词的所述查询的成本；以及

响应于确定所述查询的所述成本是最优的，重写所述查询以便使用所述等效谓词替换所述查询谓词。

18. 根据权利要求 17 的计算机系统，其中所述一个列包括生成列。

19. 根据权利要求 18 的计算机系统，其中所述操作还包括：

在规则库中查找定义所述生成列的函数的反函数；以及

使用所述反函数创建所述谓词映射器。

20. 根据权利要求 18 的计算机系统，其中所述操作还包括：

确定定义所述生成列的函数可逆；

使用数学方程式查找所述函数的反函数；以及

使用所述反函数创建所述谓词映射器。

21. 根据权利要求 20 的计算机系统，其中用于确定所述函数可逆的操作进一步包括：

确定所述函数是一对一的。

22. 根据权利要求 17 的计算机系统，其中所述谓词映射器包括模式对象，并且定义两个或更多列之间的关系。

23. 根据权利要求 17 的计算机系统，其中由用户创建所述谓词映射器。

24. 根据权利要求 17 的计算机系统，其中由数据库引擎创建所述谓词映射器。

使用谓词映射器优化查询

技术领域

[0001] 本发明的各实施例涉及使用谓词映射器优化查询。各实施例通过获得从生成列 (generated column) 到基列的逆映射而优化某些查询。

背景技术

[0002] 关系数据库管理系统 (RDBMS) 软件可以使用结构化查询语言 (SQL) 接口。RDBMS 使用关系技术在关系数据库中存储和检索数据。关系数据库被组织成表，这些表包括数据行和列。行可以称为元组或记录或行。数据库通常具有多个表，并且每个表通常具有多个记录和多个列。

[0003] 可以使用索引访问数据库中的表。索引是到表中的记录的引用（例如，指针）的有序集合。索引用于使用键（即，记录的一个字段或属性，其对应于列）访问表中的每个记录。术语“键”还可以称为“索引键”。如果没有索引，则查找记录需要扫描（例如，线性地）整个表。索引提供一种访问表中的数据的备用技术。在构建表之后，用户可以创建针对表的索引。索引基于表的一个或多个列。

[0004] 查询可以被描述为基于特定条件的对数据库中的信息的请求。查询通常包括一个或多个谓词。谓词可以被描述为表达或暗示比较运算（例如， $A = 3$ ）的搜索条件的元素。

[0005] 关系数据库表中的生成列用于存储预算值。查询优化器使用生成列替换查询中与生成列的表达式等效的表达式。但是，具有数种情况，其中生成列用于存储用户友好格式的底层数据，并且直接在应用查询中使用。例如，生成列可以是整数基列的 0 填充字符串表示。尽管整数比较快于字符串比较，但将针对 0 填充字符串表示执行字符串比较。

发明内容

[0006] 提供一种用于重写查询的计算机实现的方法、计算机程序产品和系统。接收包括查询谓词的查询。确定所述查询谓词包括具有关联的谓词映射器的一个列。使用提供从所述一个列到另一个列的映射的所述谓词映射器，针对所述查询谓词生成等效谓词。评估具有所述等效谓词的所述查询的成本。响应于确定所述查询的所述成本是最优的，重写所述查询以便使用所述等效谓词替换所述查询谓词。

附图说明

[0007] 现在将仅通过实例的方式参考附图描述本发明的优选实施例，这些附图是：

[0008] 图 1 以框图示出根据某些实施例的计算环境；

[0009] 图 2 以流程图示出根据某些实施例的用于经由自动检测创建谓词映射器的操作，图 2 包括图 2A 和 2B；

[0010] 图 3 以流程图示出根据某些实施例的用于查询转换的操作，图 3 包括图 3A 和 3B；

[0011] 图 4 示出根据某些实施例的谓词映射器的使用；

[0012] 图 5 示出根据某些实施例的实例用户标识 (UID)；

- [0013] 图 6 示出根据某些实施例的针对主 UID 表发出的实例选择语句；
- [0014] 图 7 示出根据某些实施例的输入查询和优化后的查询；
- [0015] 图 8 示出根据某些实施例的访问计划比较；
- [0016] 图 9 以框图示出根据某些实施例的可以使用的计算机体系架构。

具体实施方式

[0017] 出于示例目的给出了对本发明的不同实施例的描述，但所述描述并非旨在是穷举的或是限于所公开的实施例。在不偏离所述实施例的范围的情况下，对于所属技术领域的普通技术人员来说许多修改和变化都将是显而易见的。在此使用的术语的选择，旨在最好地解释实施例的原理、实际应用或对市场中的技术的技术改进，或者使所属技术领域的其它普通技术人员能理解在此所公开的实施例。

[0018] 从第一方面看，本发明提供一种用于重写查询的方法，包括：通过计算机的处理器接收包括查询谓词的查询；确定所述查询谓词包括具有关联的谓词映射器的一个列；使用提供从所述一个列到另一个列的映射的所述谓词映射器，针对所述查询谓词生成等效谓词；评估具有所述等效谓词的所述查询的成本；以及响应于确定所述查询的所述成本是最优的，重写所述查询以便使用所述等效谓词替换所述查询谓词。

[0019] 优选地，本发明提供一种方法，其中所述列包括生成列。

[0020] 优选地，本发明提供一种方法，还包括：在规则库中查找定义所述生成列的函数的反函数；以及使用所述反函数创建所述谓词映射器。

[0021] 优选地，本发明提供一种方法，还包括：确定定义所述生成列的函数可逆；使用数学方程式查找所述函数的反函数；以及使用所述反函数创建所述谓词映射器。

[0022] 优选地，本发明提供一种方法，其中确定所述函数可逆进一步包括：确定所述函数是一对一的。

[0023] 优选地，本发明提供一种方法，其中所述谓词映射器包括模式对象 (schema object)，并且定义两个或更多列之间的关系。

[0024] 优选地，本发明提供一种方法，其中由用户创建所述谓词映射器。

[0025] 优选地，本发明提供一种方法，其中由数据库引擎创建所述谓词映射器。

[0026] 从另一方面看，本发明提供一种用于重写查询的计算机程序产品，计算机程序产品包括：计算机可读存储介质，其中包含计算机可读程序代码，其中当由计算机的处理器执行时，所述计算机可读程序代码被配置为执行以下操作：接收包括查询谓词的查询；确定所述查询谓词包括具有关联的谓词映射器的一个列；使用提供从所述一个列到另一个列的映射的所述谓词映射器，针对所述查询谓词生成等效谓词；评估具有所述等效谓词的所述查询的成本；以及响应于确定所述查询的所述成本是最优的，重写所述查询以便使用所述等效谓词替换所述查询谓词。

[0027] 优选地，本发明提供一种计算机程序产品，其中所述一个列包括生成列。

[0028] 优选地，本发明提供一种计算机程序产品，其中当由所述计算机的所述处理器执行时，所述计算机可读程序代码被配置为执行以下操作：在规则库中查找定义所述生成列的函数的反函数；以及使用所述反函数创建所述谓词映射器。

[0029] 优选地，本发明提供一种计算机程序产品，其中当由所述计算机的所述处理器执

行时,所述计算机可读程序代码被配置为执行以下操作:确定定义所述生成列的函数可逆;使用数学方程式查找所述函数的反函数;以及使用所述反函数创建所述谓词映射器。

[0030] 优选地,本发明提供一种计算机程序产品,其中当由所述计算机的所述处理器执行时,用于确定所述函数可逆的计算机可读程序代码被配置为执行以下操作:确定所述函数是一对一的。

[0031] 优选地,本发明提供一种计算机程序产品,其中所述谓词映射器包括模式对象,并且定义两个或更多列之间的关系。

[0032] 优选地,本发明提供一种计算机程序产品,其中由用户创建所述谓词映射器。

[0033] 优选地,本发明提供一种计算机程序产品,其中由数据库引擎创建所述谓词映射器。

[0034] 从另一方面看,本发明提供一种用于重写查询的计算机系统,包括:处理器;以及耦合到所述处理器的存储设备,其中所述存储设备具有存储在其上的程序,并且其中所述处理器被配置为执行程序的指令以便执行操作,其中操作包括:接收包括查询谓词的查询;确定所述查询谓词包括具有关联的谓词映射器的一个列;使用提供从所述一个列到另一个列的映射的所述谓词映射器,针对所述查询谓词生成等效谓词;评估具有所述等效谓词的所述查询的成本;以及响应于确定所述查询的所述成本是最优的,重写所述查询以便使用所述等效谓词替换所述查询谓词。

[0035] 优选地,本发明提供一种计算机系统,其中所述一个列包括生成列。

[0036] 优选地,本发明提供一种计算机系统,其中所述操作还包括:在规则库中查找定义所述生成列的函数的反函数;以及使用所述反函数创建所述谓词映射器。

[0037] 优选地,本发明提供一种计算机系统,其中所述操作还包括:确定定义所述生成列的函数可逆;使用数学方程式查找所述函数的反函数;以及使用所述反函数创建所述谓词映射器。

[0038] 优选地,本发明提供一种计算机系统,其中用于确定所述函数可逆的操作进一步包括:确定所述函数是一对一的。

[0039] 优选地,本发明提供一种计算机系统,其中所述谓词映射器包括模式对象,并且定义两个或更多列之间的关系。

[0040] 优选地,本发明提供一种计算机系统,其中由用户创建所述谓词映射器。

[0041] 优选地,本发明提供一种计算机系统,其中由数据库引擎创建所述谓词映射器。

[0042] 图1以框图示出根据某些实施例的计算环境。服务器计算机100耦合到数据库150和规则库180。服务器计算机100包括数据库管理系统(DBMS)110。DBMS包括数据库引擎120、查询优化器130和一个或多个谓词映射器140。所述数据库存储一个或多个表160和一个或多个索引170。规则库180将函数映射到其反函数以便生成反函数。

[0043] 在某些实施例中,谓词映射器140创建从一个列到一个或多个其它列的映射。在某些实施例中,一个列和另一个列来自同一个表。在某些其它实施例中,另一个列是生成列。在某些实施例中,一个列与另一个列之间具有关系,该关系复杂并且不能表示为生成列表达式;在这些情况下,使用诸如数据库触发器之类的技术封装操作以便生成另一个列。在某些实施例中,谓词映射器140创建从生成列到一个或多个其它基列的映射。查询优化器130利用谓词映射器140重写查询。

[0044] 在某些实施例中，用户创建谓词映射器 140 以便定义从一个列到另一个列的映射。此类用户创建的谓词映射器 140 可用于数据库引擎 120。在某些备选实施例中，例如当查询谓词包括生成列时，数据库引擎 120 创建谓词映射器 140。

[0045] 图 2 以流程图示出根据某些实施例的用于经由自动检测创建谓词映射器的操作。图 2 包括图 2A 和 2B。在创建表时，数据库引擎 120 检测存在生成列并且提取定义生成列的函数。规则库 180 描述常用函数及其反向映射。针对每个生成列执行图 2 的处理。

[0046] 将智能内置于 DBMS 110 中，以便当用户提供 CREATE TABLE 语句并且 CREATE TABLE 语句具有生成列时，数据库引擎 120 尝试生成定义该生成列的函数的反函数。这将适用于数值列，其中数学函数定义生成列。因此，数据库引擎创建系统生成的谓词映射器 140。

[0047] 在图 2 中，控制在方框 200 开始，数据库引擎 120 接收具有生成列的“CREATE TABLE”数据描述语言 (DDL) 语句。各实施例将逻辑内置于数据库引擎 120 中，以便每当用户发出具有生成列定义的 CREATE TABLE DDL 语句时，数据库引擎 120 都尝试查找反函数并且使用该反函数生成谓词映射器 140。

[0048] 作为一个实例，以下 CREATE TABLE 语句具有生成列 col2。作为自动检测的一部分，DBMS 110 尝试判定 Col2 和 Col1 之间的关系是否为一对一，并且如果是，则尝试生成 Col2 的反函数并且创建谓词映射器 140。

[0049] CREATE TABLE T1

[0050] (

[0051] Col1 integer,

[0052] Col2 integer generated always as (Col1+10)/2)

[0053])

[0054] 在方框 202，数据库引擎 120 在规则库 180 中搜索定义生成列的函数 f1 的反函数。在某些实施例中，规则库 180 存储公知 / 已证明的数学函数及其反函数。如果生成列与规则库 180 中的现有函数匹配，则可以使用来自规则库本身的反函数，而不必经历生成反函数的过程。完善的函数和反函数的一些实例是：

[0055] 函数 f(x) 反函数

[0056] X+a y-a

[0057] a-x a-y

[0058] 在方框 204，数据库引擎 120 判定是否找到函数 f1 的反函数。如果是，则处理从方框 204 (图 2A) 继续到方框 210 (图 2B)，否则，处理继续到方框 206。

[0059] 在方框 206，数据库引擎 120 判定定义生成列的函数 f1 是否可逆。在某些实施例中，通过使用对判定函数 f1 是否为一对一的归纳做出该判定。如果函数 f1 可逆，则处理从方框 206 (图 2A) 继续到方框 208 (图 2B)，否则，处理完成。一对一可以描述为数学术语。并非所有的函数都具有反函数。当且仅当生成列集中的每个元素对应于基列中的唯一值时，函数才可逆。因此，在生成反函数之前，数据库引擎 120 检查定义生成列的函数是否满足该条件。这可以使用数学归纳法确定。

[0060] 在方框 208，数据库引擎 120 使用数学方程式查找函数 f1 的反函数。在方框 210，数据库引擎 120 使用函数 f1 的反函数生成谓词映射器。在方框 212，数据库引擎 120 存储谓词映射器 (例如，在系统目录中)。

[0061] 以下是如何针对定义生成列的函数 gen_col 获得反函数的一个实例,如下所示:

[0062] $\text{gen_col} = (\text{base_col}+10)/2$

[0063] 数据库引擎 120 确定函数 gen_col 可逆,因为该函数是一对一的。以下伪代码表示数据库引擎 120 以数学方式查找函数 gen_col 的反函数而执行的处理:

[0064] Rewrite the function as: $y = (x+10)/2$

[0065] Multiply both sides by the denominator: $2y = x+10$

[0066] Bring all terms involving x to the Left Hand Side (LHS) and other

[0067] terms to the Right Hand Side (RHS): $x = 2y-10$

[0068] Solving for x: $x = 2y - 10$

[0069] 然后,数据库引擎 120 确定函数 gen_col 的反函数为:

[0070] $(2*\text{gen_col}-10)$

[0071] 在其中从生成列到一个或多个基列的映射包括复杂逻辑的情况下,用户可以显式地创建谓词映射器 140。在查询转换期间,数据库引擎 120 使用该谓词映射器进行谓词替换。

[0072] 图 3 以流程图示出根据某些实施例的用于查询转换的操作。图 3 包括图 3A 和 3B。在某些实施例中,图 3 的操作表示要在查询优化的查询重写阶段期间应用的查询转换逻辑。在某些实施例中,将查询提交到数据库引擎 120 时,查询优化器 130 标识查询中包括生成列的任何谓词,并且尝试通过查询重写优化该查询。

[0073] 在图 3 中,控制在方框 300 开始,查询优化器 130 接收包括查询谓词的查询。在方框 302,查询优化器 130 判定查询谓词是否包括与谓词映射器关联的一个列。如果是,则处理继续到方框 304,否则,处理结束。所述查询谓词可以描述为原始查询谓词,并且所述一个列是原始查询谓词中的列。在某些实施例中,所述列是生成列,并且查询优化器 130 判定对于生成列是否存在谓词映射器(即,查询优化器 130 检查生成列是否具有与其关联的谓词映射器)。

[0074] 在方框 304,查询优化器 130 使用提供从原始查询谓词中的一个列到另一个列的映射的谓词映射器,针对查询谓词生成等效谓词。在某些实施例中,查询优化器 130 生成等效谓词,其中左手侧 (LHS) 是基列,并且右手侧 (RHS) 是谓词映射器返回的值。具体地说,基于谓词映射器中提供的细节,生成使用一个或多个基列的等效谓词。在方框 306,查询优化器 130 评估使用等效谓词的查询的成本。从方框 306(图 3A),处理继续到方框 308(图 3B)。在方框 308,查询优化器 130 判定成本是否是最优的。如果是,则处理继续到方框 310,否则,处理结束。查询优化器 130 是基于成本的(其中成本映射到运行查询所需的时间),并且成本包括处理成本和输入 / 输出 (I/O) 成本。在某些实施例中,使用等效谓词的查询的最优成本指以下成本:小于不使用等效谓词的查询(即,原始查询)的成本。如果查询优化器 130 使用函数映射器以等效谓词替换原始谓词,并且具有等效谓词的成本更小,则使用等效谓词。

[0075] 在方框 310,查询优化器 130 重写查询以便使用等效谓词替换查询谓词。在方框 312,数据库引擎 120 执行重写后的查询。

[0076] 查询优化器 130 和谓词映射器 140 应用于广泛的优化情况,包括使用更有效数据

类型的比较（数值与字符串）、分解谓词以便限制扫描范围，以及索引使用。查询优化器 130 执行从生成列到一个或多个基列的反向映射。因为映射反向，所以数据库引擎 120 中存在尝试将生成列隐式地映射到一个或多个基列的处理逻辑。

[0077] 仅为了解，在此提供各种实例。

[0078] 在客户情况 1 中，唯一 ID 是整数数据，但唯一 ID 表示为使用 0 填充的字符串以便具有统一、固定长度的表示。字符串表示是基于整数值的生成列。字符串表示对于应用 / 用户而言已知并且在查询中使用。但是，在该实例中，客户想要在查询处理期间使用整数列以实现性能优势。以下是具有实例查询和重写后的查询的实例表“TAB1”：

[0079] 实例表：

[0080] TAB1，其具有列 uid 整数，并且 Unique_id 是 varchar 列，其生成为

[0081] Unique_Id = lpad(uid, 6, '0')

[0082] 实例查询：

[0083] select*from tab1 where Unique_Id = '000012011'

[0084] 重写后的查询：

[0085] select*from tab1 where uid = 12011

[0086] 在客户情况 2 中，整数值以十六进制（“hex”）值格式使用，存储为字符串。应用在查询中使用 hex 格式，但是整数比较产生更好的性能结果。以下是具有实例查询和重写后的查询的实例表“TAB2”：

[0087] 实例表：

[0088] TAB2，其具有列 cid 整数，并且 CUST_ID 是 varchar 列，其生成为

[0089] CUST_ID = hex(cid)

[0090] 实例查询：

[0091] select*from tab2 where CUST_ID = 0x'94A'

[0092] 重写后的查询：

[0093] select*from tab2 where cid = 2378

[0094] 客户情况 3 是银行实例，其中客户的唯一账户标识符（ID）包括数个组成部分（例如区域代码、分行代码和序列号）的组合并且表示为生成列。当客户数量非常大，并且不同区域和分行数量相比之下非常小时，将用于 account_id 上的谓词分解成用于各组成部分上的谓词涉及扫描更少的页并且提高查询性能。以下是具有实例查询和重写后的查询的实例表“TAB3”：

[0095] 实例表：

[0096] TAB3，其具有列 rc char(2)、bc char(3)、s1no 整数，并且 ACCOUNT_ID 是 varchar 列，其生成为 ACCOUNT_ID = concat(rc, bc, s1no)

[0097] 实例查询：

[0098] select*from tab3where ACCOUNT_ID = '112221234'

[0099] 重写后的查询：

[0100] select*from tab3where rc = '11' and bc = '222' and s1no = 1234

[0101] 在客户情况 4 中，将基列上的复合函数的结果存储为生成列，并且在基列上定义索引。在此，如果用户提供具有生成列的谓词，则各实施例使用基列并且利用在基列上定义

的索引来重写谓词。以下是具有实例查询和重写后的查询的实例表“TAB4”：

[0102] 实例表：

[0103] TAB4, 具有在其上包含索引的 sales 列以及生成为 $0.2 * \text{sales} + 10000$ 的 BONUS 列

[0104] 实例查询：

[0105] select*from tab4 where BONUS>20000

[0106] 重写后的查询：

[0107] select*from tab4 where sales>50000

[0108] 谓词映射器 140 可以被描述为将表中的生成列映射到一个或多个基列的新模式对象。谓词映射器可以由数据库引擎 120 生成, 或者可以由用户使用数据描述语言 (DDL) 语句显式定义。谓词映射器 140 可以用于定义任何列之间的关系 (例如, 基于生成列的关系)。

[0109] 某些实施例使用以下语法创建谓词映射器：

[0110] CREATE PREDICATE MAPPER UIDPMap

[0111] FOR uid TO fflag, nuid ON uid_master

[0112] RETURN NEWRHS =

[0113] (substr(OLDRHS, 1, 1), INT(substr(OLDRHS, 2, 11)))

[0114] 图 4 示出根据某些实施例的谓词映射器的使用。使用通过创建语句 410 创建的谓词映射器, 将原始查询 400 重写为查询 420。

[0115] 某些实施例使用以下语法创建用户定义的谓词映射器：

[0116] CREATE PREDICATE MAPPER

[0117] <name>

[0118] FOR<src_gen_col>

[0119] [TO<base_col1>, <base_col2>, <base_col3>, ...]

[0120] ON<table_name>

[0121] RETURN NEWRHS = (function1(OLDRHS),

[0122] [function2(OLDRHS), function3(OLDRHS), ...]);

[0123] 其中：

[0124] <name> 标识谓词映射器对象

[0125] <src_gen_col> 是生成列的名称

[0126] <base_col1>, <base_col2>, <base_col3>, ... 是在其上定义生成列的一个或多个基列的列表。这是可选的。如果未提供, 则数据库引擎 120 按照一个或多个基列的列表在生成列定义中出现的顺序隐式地获得该列表。

[0127] <table_name> 是具有生成列的表

[0128] NEWRHS 是将包含新谓词的 RHS 值的关键字

[0129] OLDRHS 是将包含旧谓词的 RHS 值的关键字

[0130] function1、function2、function3... 是内置或用户定义的函数

[0131] 以下是创建客户情况 1 的谓词映射器的一个实例, 其中使用内置函数 INT 查找新谓词的 RHS :

[0132] CREATE PREDICATE MAPPER pmap1FOR unique_id ON tab1

[0133] RETURN NEWRHS = INT(OLDRHS)

[0134] 使用该谓词映射器定义,数据库引擎 120 使用谓词 uid = 12210 替换谓词 unique_id = '000012210'。

[0135] 以下是创建客户情况 2 的谓词映射器的一个实例,其中使用用户定义的函数 hex_to_dec 查找新谓词的 RHS :

[0136] CREATE PREDICATE MAPPER pmap2FOR cust_id ON tab2

[0137] RETURN NEWRHS = hex_to_dec(OLDRHS)

[0138] 使用该谓词映射器,数据库引擎 120 使用谓词 cid = 9802 替换谓词 cust_id = x'264A'。

[0139] 以下是创建客户情况 3 的谓词映射器的一个实例,其中使用内置函数 substr 返回值列表,每个基列一个值 :

[0140] CREATE PREDICATE MAPPER pmap3FOR account_id TO rc,

[0141] bc, slno ON tab3

[0142] RETURN NEWRHS = (substr(OLDRHS, 1, 2),

[0143] substr(OLDRHS, 3, 3), substr(OLDRHS, 6, 4))

[0144] 使用该谓词映射器,数据库引擎 120 使用谓词 rc = '11', bc = '222' 以及 slno = '1234' 替换谓词 account_id = " 112221234"。

[0145] 某些实施例适用于用户标识 (UID), 其中 UID 是不包括字母字符的数字 (因为特定字母依赖于特定脚本 / 语言和文化)。在某些实施例中,针对 UID 使用 12 个十进制数字。版本号 :a1 = 0 表示个体,而 a1 = 1 表示实体等。如果使用 2-9 数字 (a1 = 2,3…9) 指定 UID, 则具有 800 亿个数字。UID 可以是使用 0 填充的数字,以使 a2…a11 具有固定长度。在某些实施例中,UID 认证是具有严格响应时间要求的操作。图 6 示出根据某些实施例的针对主 UID 表 610 发出的实例选择语句 600。在主 UID 表 610 中,UID 是 varchar。图 7 示出根据某些实施例的输入查询 600 和优化后的查询 700。图 8 示出根据某些实施例的访问计划比较。具体地说,将输入查询 600 的访问计划 800 与优化后的查询 700 的访问计划 810 相比较,以便显示成本改善了 25%。较低的成本意味着较快的查询执行时间。

[0146] 各实施例扩展 SQL 以便支持称为谓词映射器的提供从生成列到基列的映射的新模式对象。此外,使用各实施例,查询优化器 130 重写查询以便实现更好的性能。

[0147] 各实施例通过在数据库中获得逆映射来优化查询,方法是 : 扩展 SQL 以便创建称为谓词映射器的新模式对象,从而提供从生成列到基列的映射。各实施例接收对数据库引擎 120 的查询并判定查询中的谓词是否包括生成列,并且如果是,则使用基列 (多个) 和与该列关联的谓词映射器中提供的细节生成等效谓词。各实施例评估使用等效谓词的查询的成本,并且在查询重写阶段期间利用谓词映射器以便生成备选谓词,评估备选谓词的成本,以及基于该成本确定性能最好的查询执行计划。各实施例经由自动检测并基于与列关联的反函数来生成谓词映射器。

[0148] 各实施例重写查询以便利用生成列。各实施例使用 DDL 从用户处获得列之间的关系,并且让查询优化器 130 使用该信息产生更好的查询计划。各实施例使得查询处理器能够知道在基列上使用备选但等效谓词可以被证明更有效。

[0149] 所属技术领域的技术人员知道,本发明的各个方面可以实现为系统、方法或计算

机程序产品。因此,本发明的各个方面可以具体实现为以下形式,即:完全的硬件实施方式、完全的软件实施方式(包括固件、驻留软件、微代码等),或硬件和软件方面结合的实施方式,这里可以统称为“电路”、“模块”或“系统”。此外,本发明的各个方面还可以实现为在一个或多个计算机可读介质中的计算机程序产品的形式,该计算机可读介质中包含计算机可读的程序代码。

[0150] 可以采用一个或多个计算机可读介质的任意组合。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一电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者上述的任意合适的组合。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、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、固态存储器、磁带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。

[0151] 计算机可读的信号介质可以包括例如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,包括但不限于一电磁信号、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。

[0152] 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、有线、光缆、RF等等,或者上述的任意合适的组合。

[0153] 可以以一种或多种程序设计语言的任意组合来编写用于执行本发明的各个方面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++等,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)。

[0154] 下面将参照根据本发明实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述本发明实施例的各个方面。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机程序指令实现。这些计算机程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器,从而生产出一种机器,使得这些指令在通过计算机或其它可编程数据处理装置的处理器执行时,产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。

[0155] 也可以把这些计算机程序指令存储在计算机可读介质中,这些指令使得计算机、其它可编程数据处理装置、或其它设备以特定方式工作,从而,存储在计算机可读介质中的指令就产生出包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的指令的制造品(article of manufacture)。

[0156] 也可以把计算机程序指令加载到计算机、其它可编程数据处理装置、或其它设备上，使得在计算机、其它可编程装置或其它设备上执行一系列操作处理（例如，操作或步骤），以产生计算机实现的过程，从而使得在计算机或其它可编程装置上执行的指令提供实现流程图和 / 或框图中的一个或多个方框中规定的功能 / 动作的过程。

[0157] 实现所述操作的代码还可以以硬件逻辑或电路（例如，集成电路芯片、可编程门阵列（PGA）、专用集成电路（ASIC）等）实现。硬件逻辑可以耦合到处理器以便执行操作。

[0158] 相互通信的设备不必持续地相互通信，除非另外明确指出。此外，相互通信的设备可以直接或通过一个或多个媒介间接地通信。

[0159] 具有若干相互通信的组件的实施例的描述并非暗示所有此类组件都是必需的。相反，描述了多种可选的组件以示本发明的更多可能实施例。

[0160] 此外，尽管可以按照连续的顺序来描述过程步骤、方法步骤、算法等，但是此类过程、方法和算法可以被配置为以替代顺序工作。换句话说，可以描述的步骤的任何序列或顺序并不一定指示要求按此顺序执行步骤。可以按可实现的任何顺序执行在此描述的过程的步骤。此外，可以同时执行某些步骤。

[0161] 当在此描述单个设备或物品时，将显而易见的是，可以使用多个设备 / 物品（无论它们是否协作）来代替单个设备 / 物品。同样，当在此描述多个设备或物品（无论它们是否协作）的情况下，将显而易见的是，可以使用单个设备 / 物品来代替多个设备或物品，或者可以使用不同数量的设备 / 物品来代替所示数量的设备或程序。设备的功能和 / 或特性可以备选地由一个或多个其它未明确描述为具有此类功能 / 特性的设备来体现。因此，本发明的其它实施例不必包括设备本身。

[0162] 流程图的示出的操作显示某些事件以特定的顺序发生。在各备选实施例中，某些操作可以以不同的顺序被执行、修改或删除。此外，可以向上述逻辑添加操作并仍然符合所描述的实施例。此外，此处所述的操作可以顺序地发生或者某些操作可以被并行地处理。更进一步，操作可以由单个处理单元或由分布式处理单元来执行。

[0163] 在此使用的术语只是为了描述特定的实施例并且并非旨在作为本发明的限制。如在此使用的，单数形式“一”、“一个”和“该”旨在同样包括复数形式，除非上下文明确地另有所指。还将理解，当在此说明书中使用时，术语“包括”和 / 或“包含”指定了声明的特性、整数、步骤、操作、元素和 / 或组件的存在，但是并不排除一个或多个其它特性、整数、步骤、操作、元素、组件和 / 或其组合的存在或增加。

[0164] 术语“一个实施例”、“实施例”、“多个实施例”、“所述实施例”、“所述多个实施例”、“一个或多个实施例”、“某些实施例”和“某一实施例”指“本发明（多个）的一个或多个（但不是所有）实施例”，除非另外明确指出。

[0165] 术语“包含”、“包括”、“具有”及其变型指“包括但不限于”，除非另外明确指出。

[0166] 列举的项目的列表并非暗示任何或所有的项目互相排斥，除非另外明确指出。

[0167] 下面权利要求中的对应结构、材料、操作以及所有功能性限定的装置或步骤的等同替换，旨在包括任何用于与在权利要求中具体指出的其它元件相组合地执行该功能的结构、材料或操作。出于示例和说明目的给出了对本发明实施例的描述，但所述描述并非旨在是穷举的或是将本发明限于所公开的形式。在不偏离本发明的范围和精神的情况下，对于所属技术领域的普通技术人员来说许多修改和变化都将是显而易见的。实施例的选择和描

述是为了最优化地解释本发明的原理和实际应用，并且当适合于所构想的特定使用时，使得所属技术领域的其它普通技术人员能够理解本发明的具有各种修改的各种实施例。

[0168] 图 9 示出根据某些实施例的可以使用的计算机体系架构 900。服务器计算机 100 可以实现计算机体系架构 900。计算机体系架构 900 适合于存储和 / 或执行程序代码，并且包括至少一个通过系统总线 920 直接或间接连接到存储元件 904 的处理器 902。存储元件 904 可以包括在程序代码的实际执行期间采用的本地存储器、大容量存储装置以及提供至少某些程序代码的临时存储以减少必须在执行期间从大容量存储装置检索代码的次数的高速缓冲存储器。存储元件 904 包括操作系统 905 和一个或多个计算机程序 906。

[0169] 输入 / 输出 (I/O) 设备 912、914(包括但不限于键盘、显示器、指点设备等) 可以直接或通过中间 I/O 控制器 910 与系统相连。

[0170] 网络适配器 908 也可以被连接到系统以使数据处理系统能够通过中间专用或公共网络变得与其它数据处理系统或远程打印机或存储设备相连。调制解调器、电缆调制解调器和以太网卡只是当前可用类型的网络适配器 908 中的少数几种。

[0171] 计算机体系架构 900 可以连接到存储装置 916(例如，任何类型的存储设备；非易失性存储区域，例如磁盘驱动器、光盘驱动器、磁带驱动器等)。存储装置 916 可以包括内部存储设备或附加的或网络可访问存储装置。存储装置 916 中的计算机程序 906 可以加载到存储元件 904 中，并且由处理器 902 以所属技术领域已知的方式执行。

[0172] 计算机体系架构 900 可以包括少于所示的组件、未在此示出的其它组件，或者所示组件和其它组件的某种组合。计算机体系架构 900 可以包括所属技术领域已知的任何计算设备，例如大型机、服务器、个人计算机、工作站、膝上型计算机、手持式计算机、电话设备、网络设备、虚拟化设备、存储控制器等。

[0173] 附图中的流程图和框图显示了根据本发明的不同实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分，所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现中，方框中所标注的功能可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和 / 或流程图中的每个方框、以及框图和 / 或流程图中的方框的组合，可以用执行规定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0174] 出于示例和说明目的给出了本发明实施例的上述描述。其并非旨在是穷举的或将实施例限于所公开的精确形式。根据上述教导，许多修改和变化都是可能的。其旨在实施例的范围并非由此详细描述来限制，而是由此后所附的权利要求来限制。以上说明、实例和数据提供了对实施例的组成部分的制造和使用的完整描述。由于可以在不偏离本发明的精神和范围的情况下做出许多实施例，所以实施例存在于此后所附的权利要求或随后提交的任何权利要求或其等效物之内。

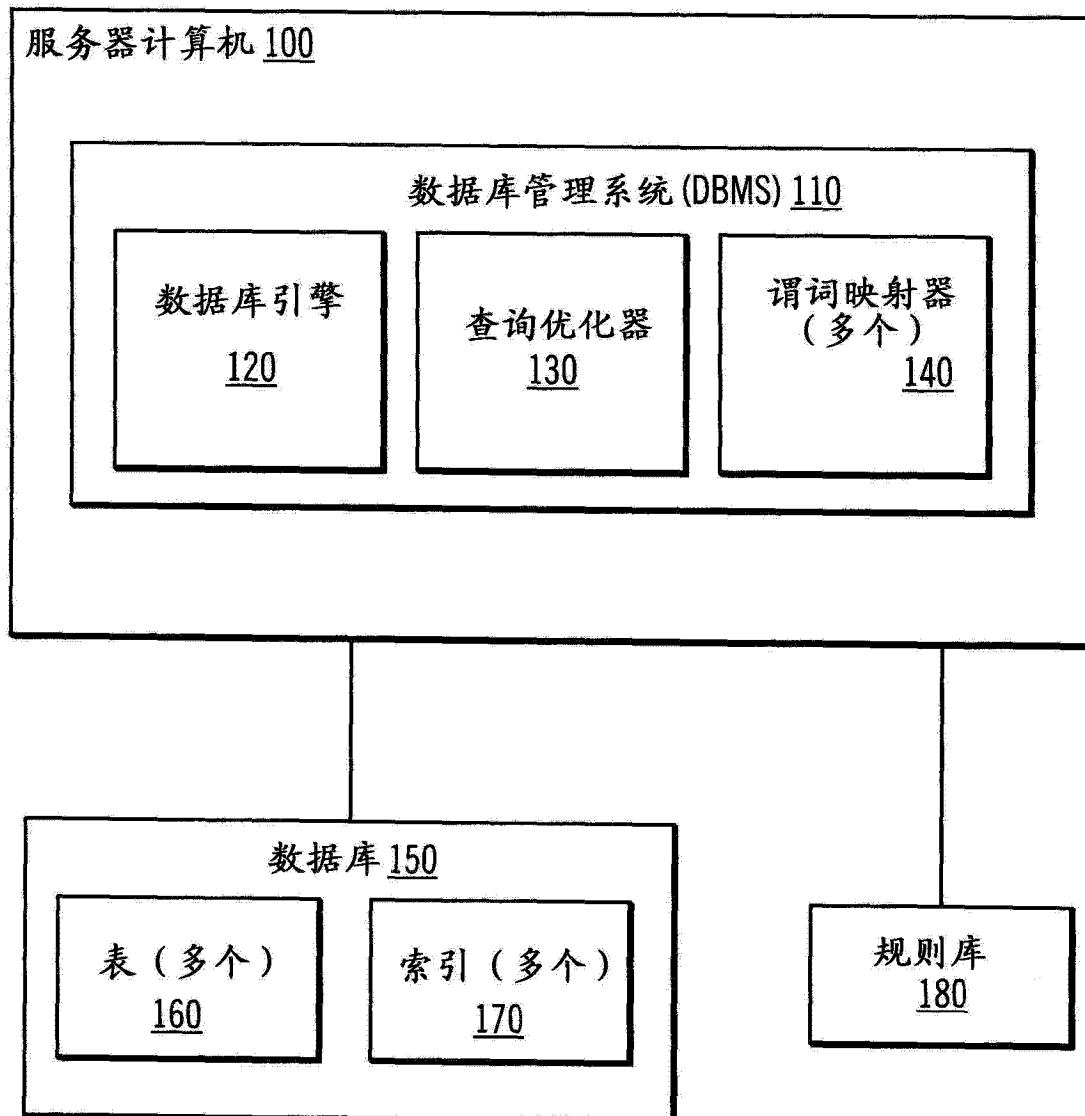


图 1

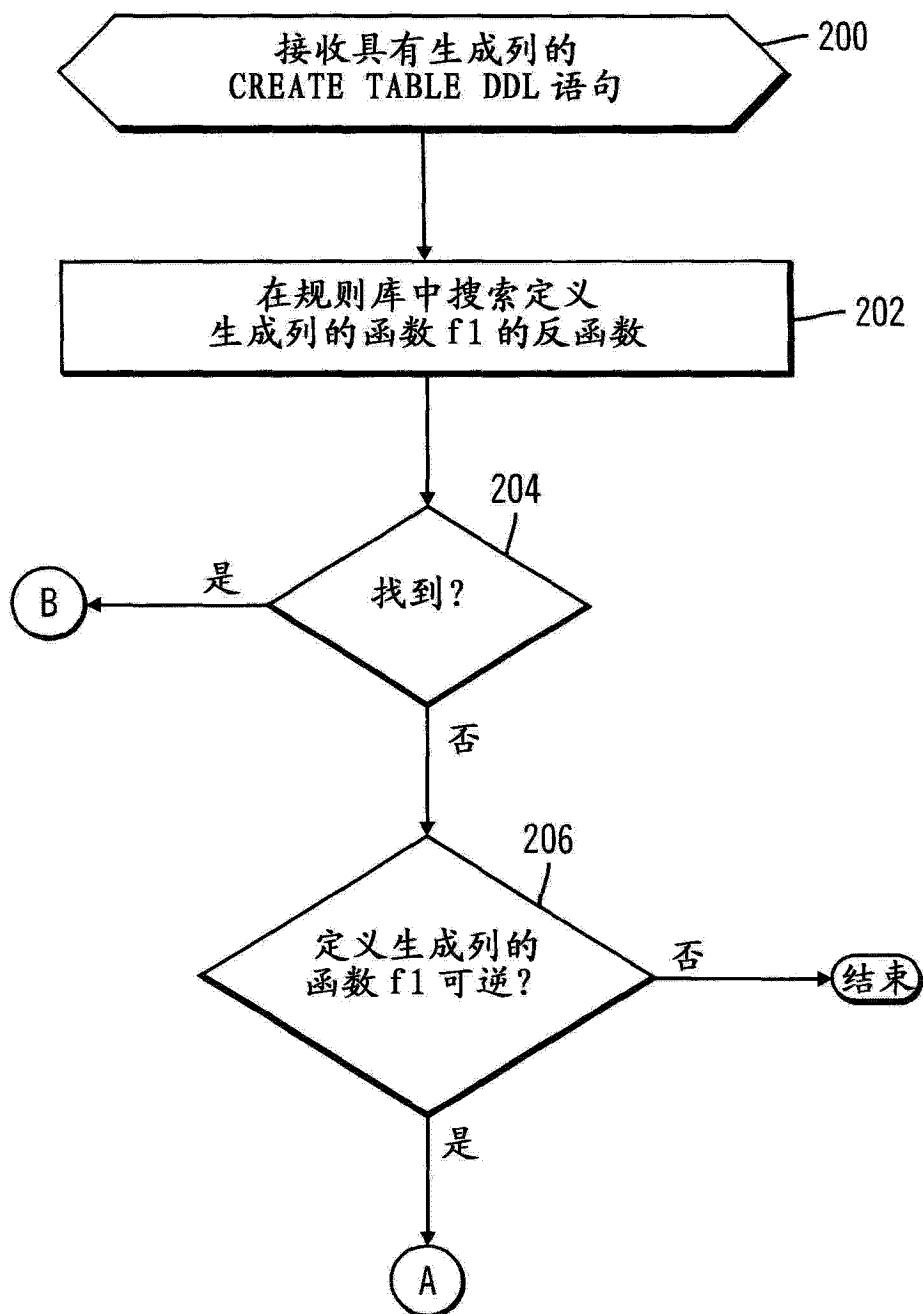


图 2A

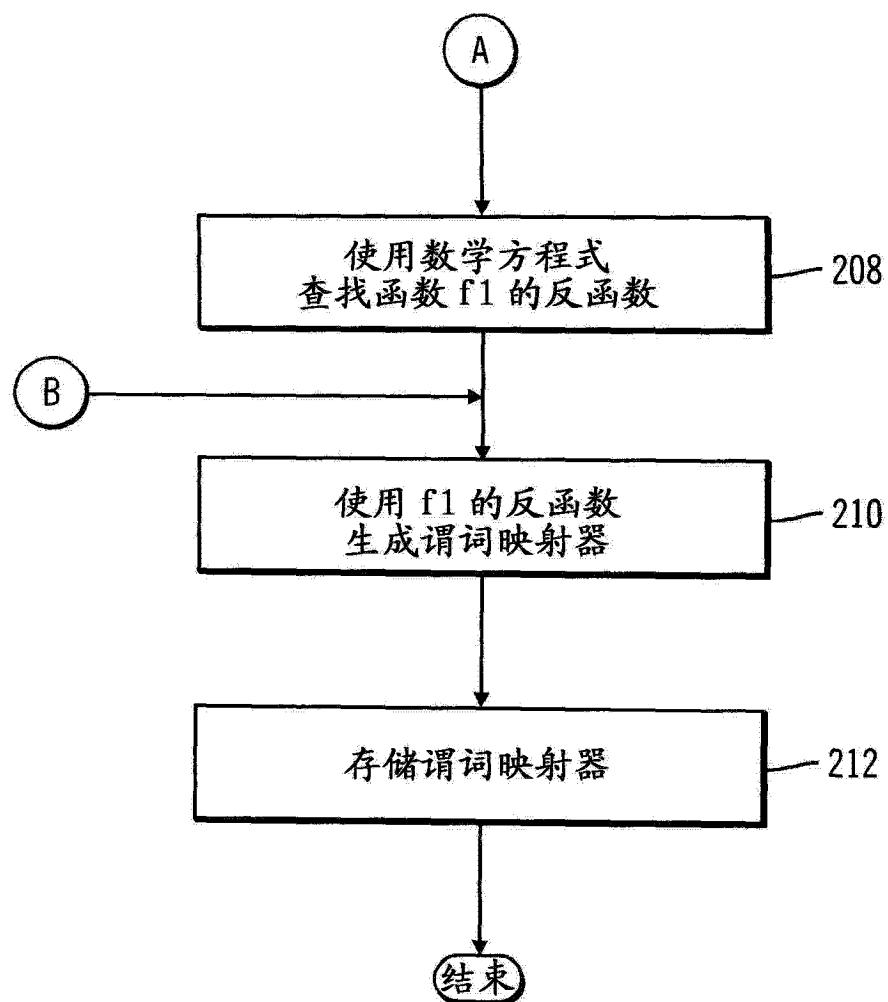


图 2B

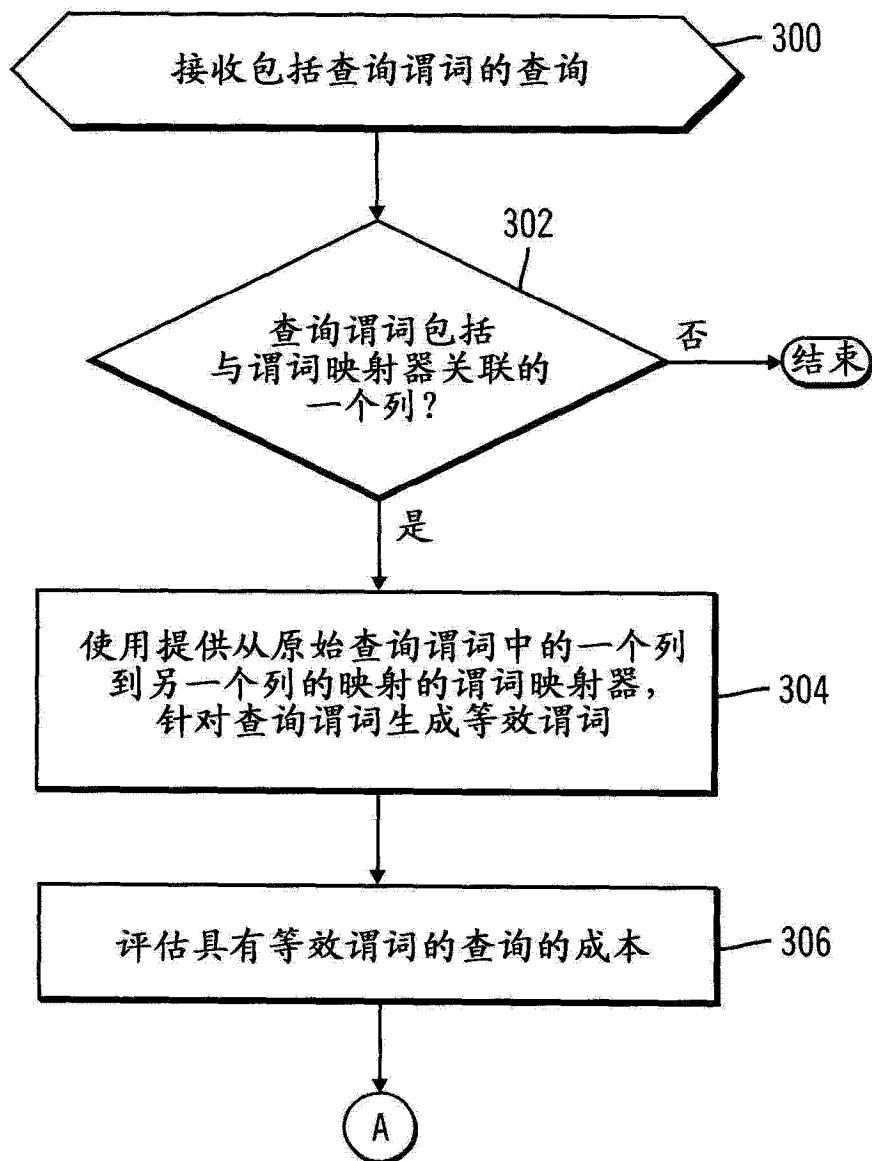


图 3A

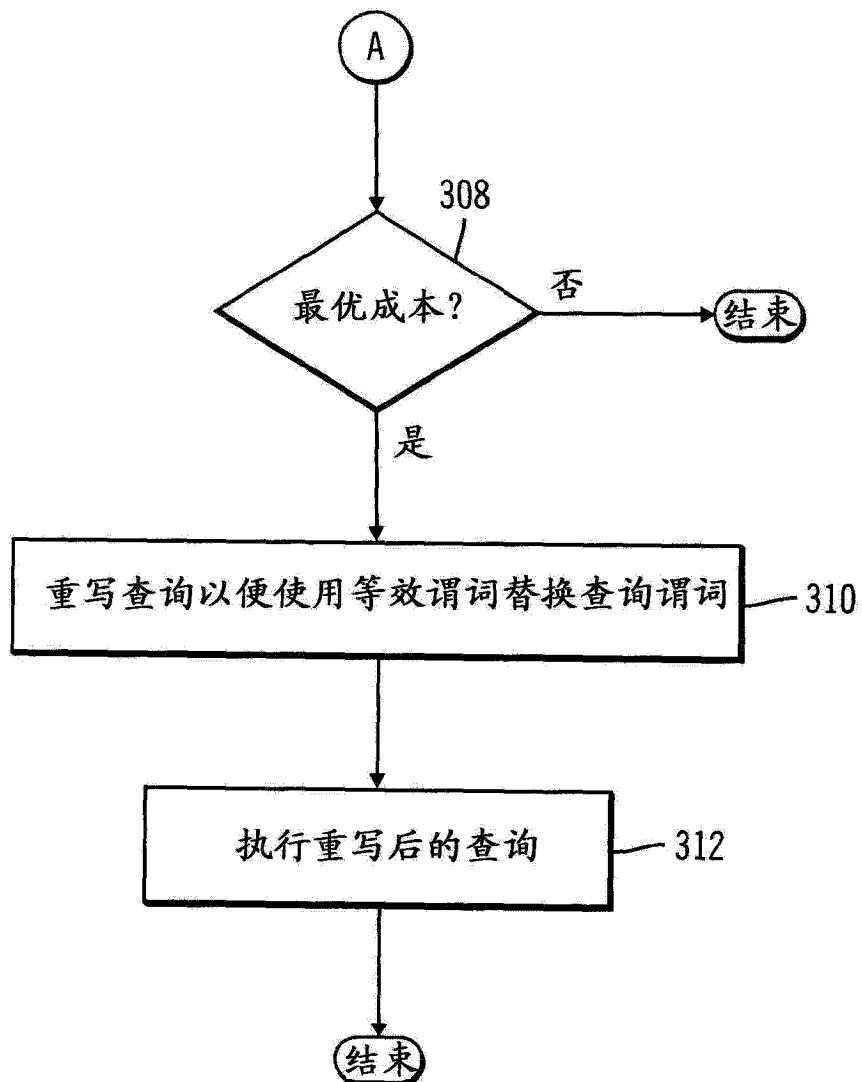


图 3B

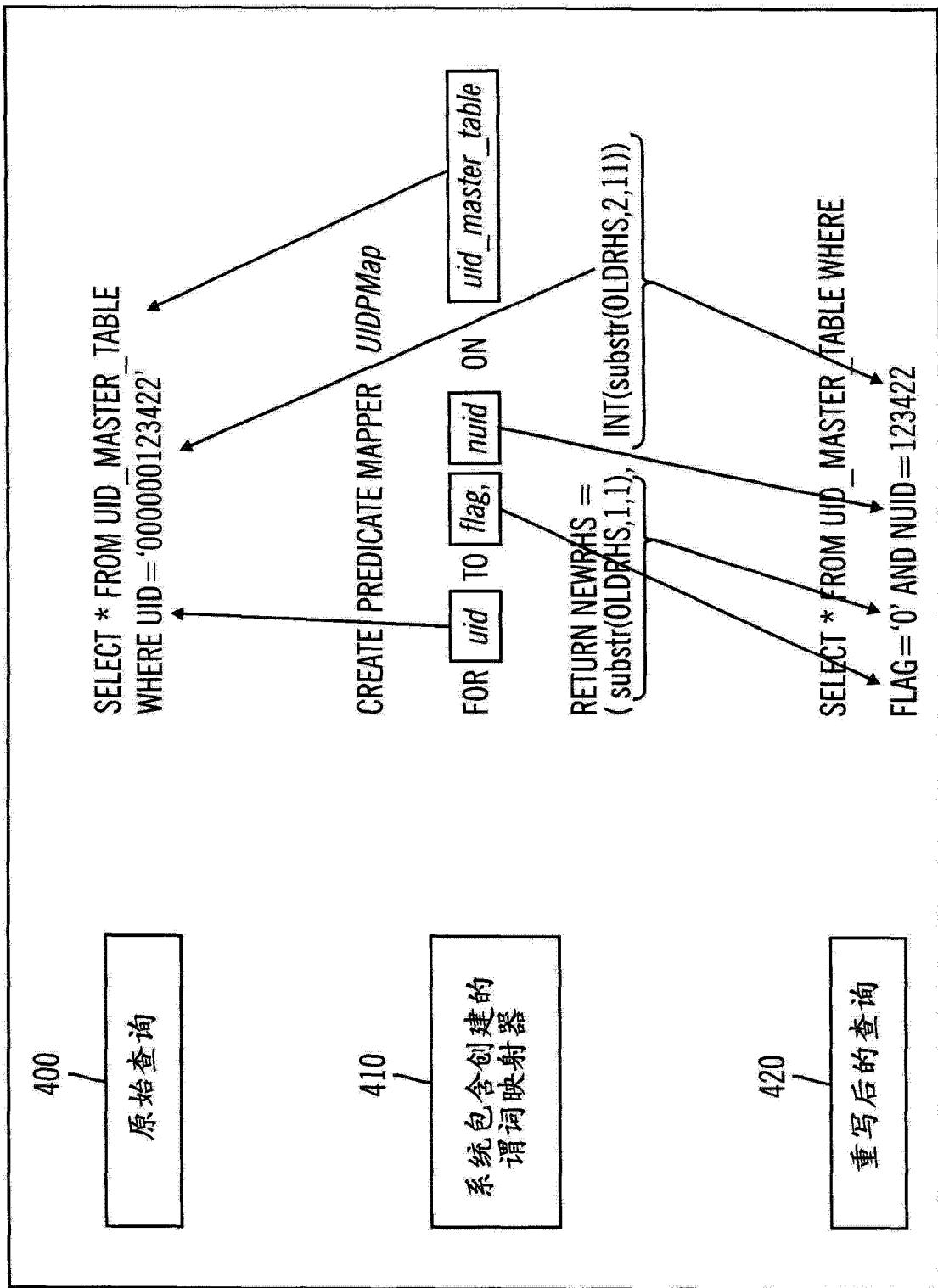


图 4

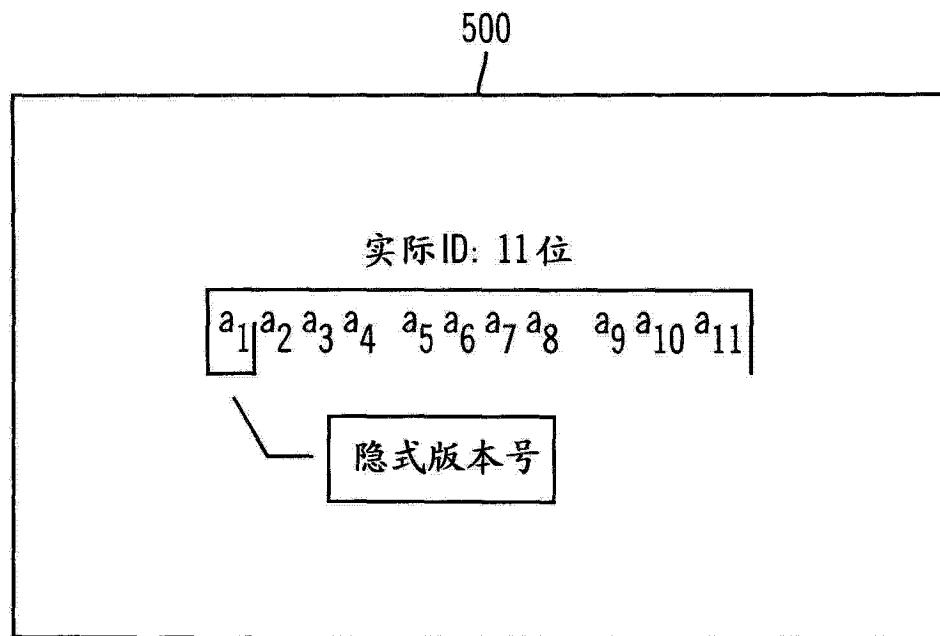


图 5

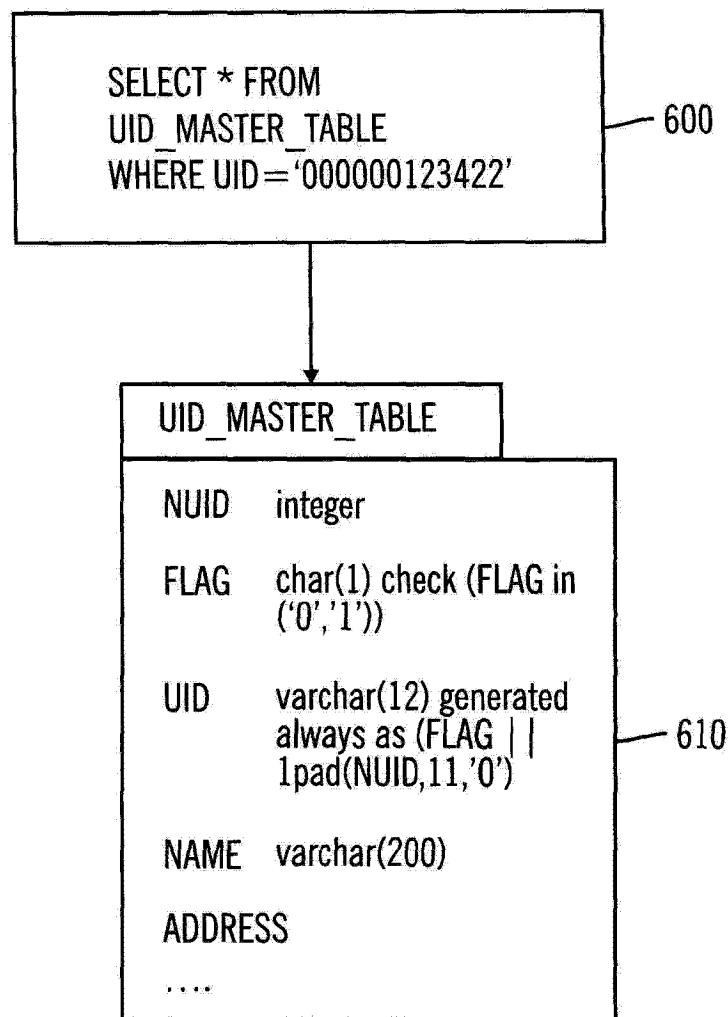


图 6

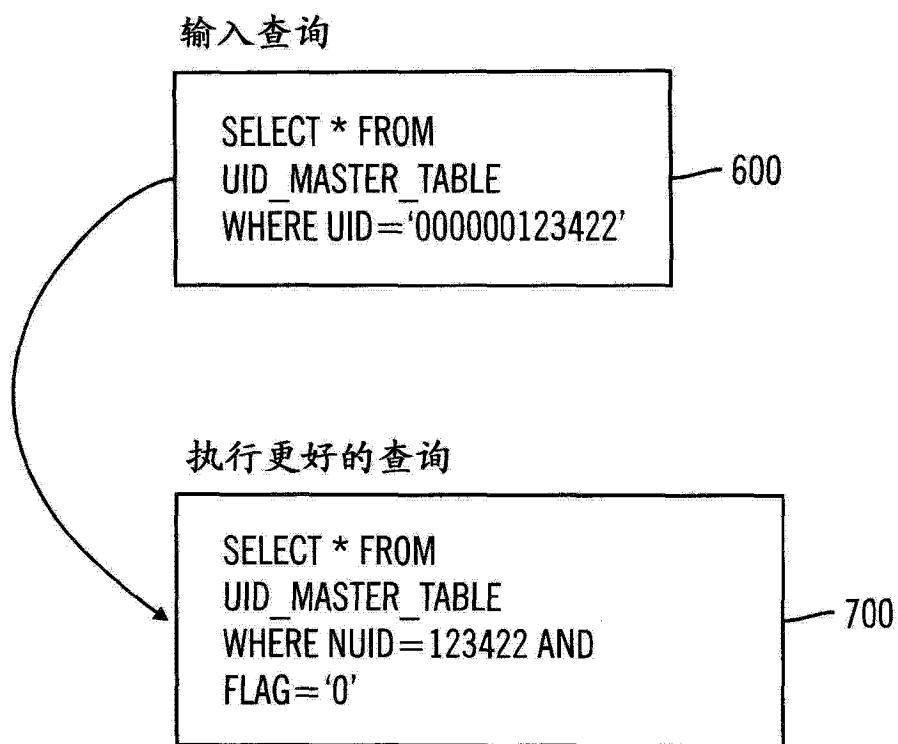
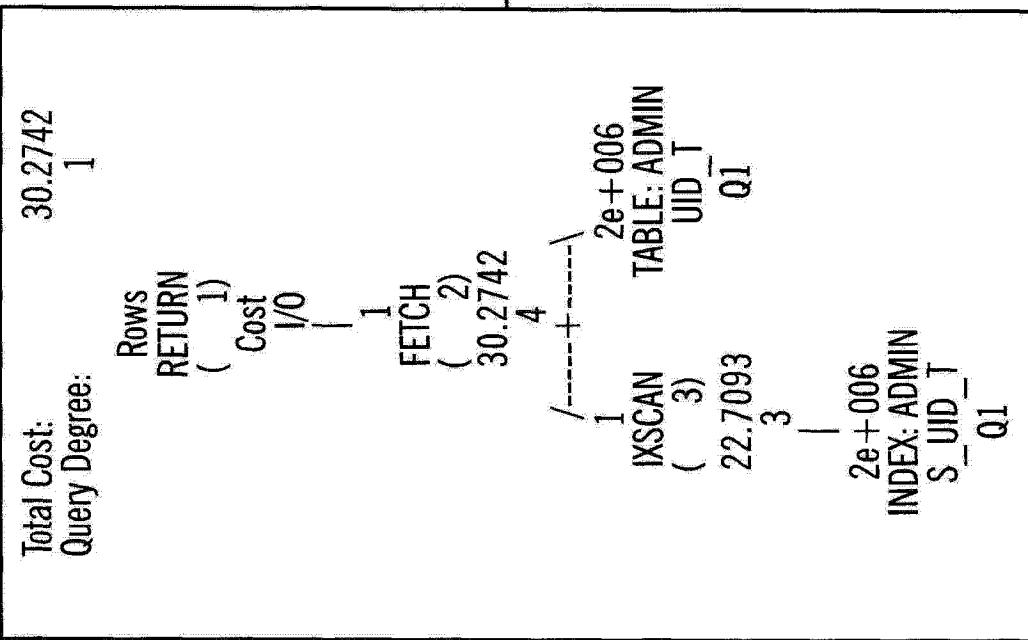
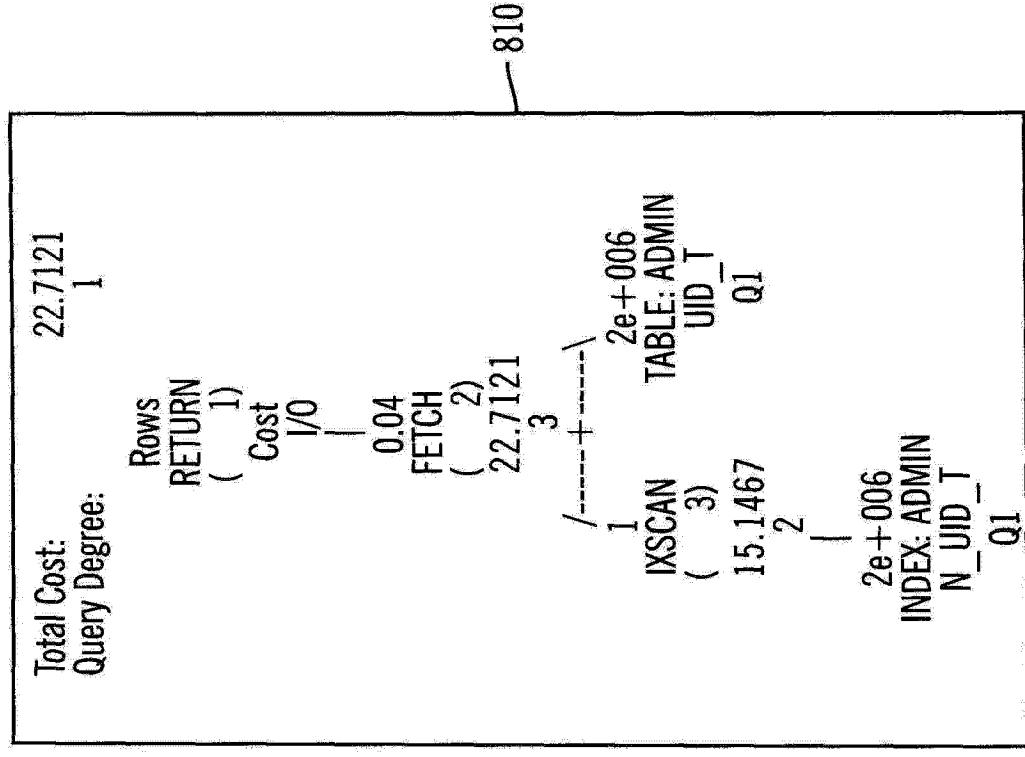


图 7

select nuid from uid_t where UID=?



select uid from uid_t where t where
FLAG = substr(?,1,1) and NUID= int(substr(?,2,11))



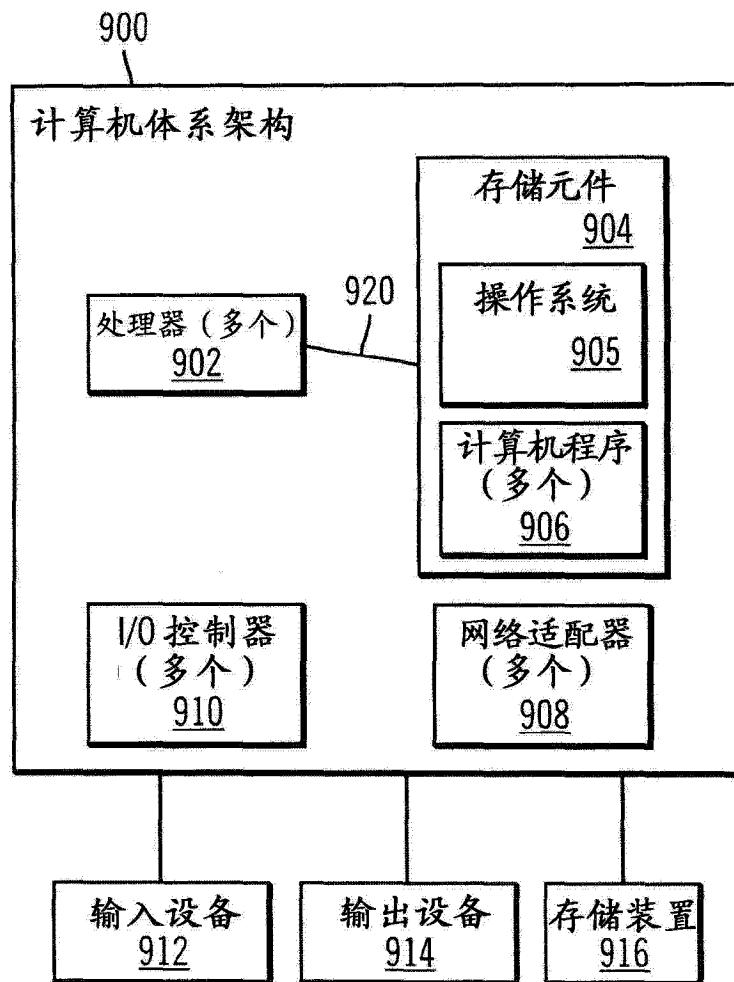


图 9