

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2018 年 12 月 20 日 (20.12.2018)



(10) 国际公布号
WO 2018/228322 A1

(51) 国际专利分类号:
G06F 17/30 (2006.01)

(21) 国际申请号: PCT/CN2018/090610

(22) 国际申请日: 2018 年 6 月 11 日 (11.06.2018)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201710447344.0 2017 年 6 月 14 日 (14.06.2017) CN

(71) 申请人: 腾讯科技(深圳)有限公司 (TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED) [CN/CN]; 中国广东省深圳市南山区高新区科技中一路腾讯大厦 35 层, Guangdong 518057 (CN)。

(72) 发明人: 李海翔 (LI, Haixiang); 中国广东省深圳市南山区高新区科技中一路腾讯大厦 35 层, Guangdong 518057 (CN)。

(74) 代理人: 北京德琦知识产权代理有限公司 (DEQI INTELLECTUAL PROPERTY LAW CORPORATION); 中国北京市海淀区知春路 1 号学院国际大厦 7 层, Beijing 100083 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,

(54) Title: OPTIMIZATION METHOD AND DEVICE FOR NON-CORRELATED SUB-QUERY AND STORAGE MEDIUM

(54) 发明名称: 一种非相关子查询的优化方法、装置和存储介质

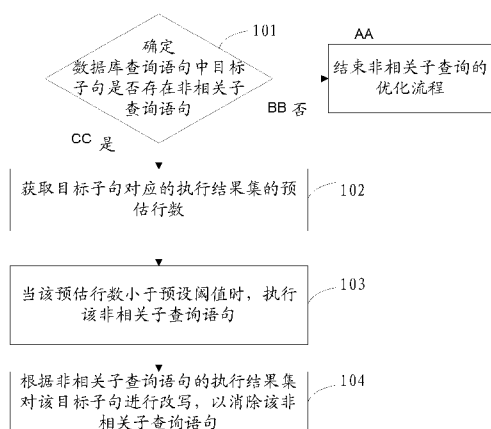


图 1b

- 101 Determine whether a non-correlated sub-query statement exists in a target sub-statement in database query statements or not
102 Obtain an estimated number of rows of an execution result set corresponding to the target sub-statement
103 Execute the non-correlated sub-query statement if the estimated number of rows is less than a preset threshold
104 Rewrite the target sub-statement according to the execution result set of the non-correlated sub-query statement to eliminate the non-correlated sub-query statement
AA End the optimization process for non-correlated sub-query
BB No
CC Yes

(57) Abstract: Embodiments of the present application provide an optimization method and device for non-correlated sub-query and a storage medium. The method comprises: determining whether a non-correlated sub-query statement exists in a target sub-statement in database query statements or not; if yes, obtaining an estimated number of rows of an execution result set corresponding to the target sub-statement; executing the non-correlated sub-query statement if the estimated number of rows is less than a preset threshold; and rewriting the target sub-statement according to the execution result set of the non-correlated sub-query statement to eliminate the non-correlated sub-query statement.

(57) 摘要: 本申请实施例公开了一种非相关子查询的优化方法、装置和存储介质。该方法包括: 确定数据库查询语句中目标子句是否存在非相关子查询语句; 若存在, 则获取所述目标子句对应的执行结果集的预估行数; 当所述预估行数小于预设阈值时, 执行所述非相关子查询语句; 根据所述非相关子查询语句的执行结果集对所述目标子句进行改写, 以消除所述非相关子查询语句。

IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

一种非相关子查询的优化方法、装置和存储介质

本申请要求于 2017 年 6 月 14 日提交中国专利局、申请号为 201710447344.0，申请名称为“一种非相关子查询的优化方法和装置”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，具体涉及一种非相关子查询的优化方法、装置和存储介质。

背景技术

数据处理系统如数据库系统、大数据处理系统等通常都以支持 SQL（Structured Query Language，结构化查询语言）功能为主要实现目标，以为用户提供数据操作便利。

在 SQL 处理过程中，需要 SQL 引擎对 SQL 语句进行词法解析、语法解析、语义分析、逻辑优化和物理优化，然后把得到的执行计划交给执行器去执行，这个过程被划分为三个阶段：分析阶段、优化阶段、执行阶段。

其中，子查询的优化发生在逻辑优化阶段，是逻辑优化阶段中的优化子阶段（通常称为查询重写）。一般来说，子查询可以分为相关子查询和非相关子查询两种类型。

发明内容

本申请实施例提供一种非相关子查询的优化方法、装置和存储介质，可以提高具有非相关子查询的数据库查询语句的执行效率。

本申请实施例提供一种非相关子查询的优化方法，应用于计算设备，包括：

确定数据库查询语句中目标子句是否存在非相关子查询语句；

若存在，则获取所述目标子句对应的执行结果集的预估行数；
当所述预估行数小于预设阈值时，执行所述非相关子查询语句；
根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，以消除所述非相关子查询语句。

本申请实施例还提供一种非相关子查询的优化装置，包括：

处理器以及与所述处理器相连接的存储器，所述存储器中存储有可由所述处理器执行的机器可读指令，所述处理器执行所述机器可读指令完成以下操作：

确定数据库查询语句中目标子句是否存在非相关子查询语句；

当确定存在非相关子查询语句时，获取目标子句对应的执行结果集的预估行数；

当所述预估行数小于预设阈值时，执行所述非相关子查询语句；

根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，以消除所述非相关子查询语句。

本申请实施例提供一种非易失性计算机可读存储介质，所述存储介质中存储有机器可读指令，所述机器可读指令可以由处理器执行以完成以下操作：

确定数据库查询语句中目标子句是否存在非相关子查询语句；

若存在，则获取所述目标子句对应的执行结果集的预估行数；

当所述预估行数小于预设阈值时，执行所述非相关子查询语句；

根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，以消除所述非相关子查询语句。

附图说明

为了更清楚地说明本申请实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域技术人员来讲，在不付出创造

性劳动的前提下，还可以根据这些附图获得其他的附图。

图1a是本申请实施例提供的非相关子查询的优化方法的实施环境示意图；

图1b是本申请实施例提供的非相关子查询的优化方法的流程示意图；

图1c是本申请实施例提供的半连接优化方式的示意图；

图1d是本申请实施例提供的物化优化方式的示意图；

图2a是本申请实施例提供的非相关子查询的优化方法的另一流程示意图；

图2b是本申请实施例提供的SQL语句处理的流程示意图；

图3a是本申请实施例提供的非相关子查询的优化装置的第一种结构示意图；

图3b是本申请实施例提供的非相关子查询的优化装置的第二种结构示意图；

图3c是本申请实施例提供的非相关子查询的优化装置的第三种结构示意图；

图4是本申请实施例提供的服务器的结构示意图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅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，本领域技术人员在没有作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。

目前，非相关子查询的常规优化方式主要包括半连接优化和“物化”优化，其中，半连接优化，属于子查询上拉技术，是用半连接(semi join)的语义消除子查询的一种优化形式，属于逻辑优化子阶段；物化优化是把非相关子查询子句标识为“可物化”，并在执行阶段执行一次，然后

把结果缓冲，供后续使用。

然而，采用上述非相关子查询的优化方式对具有非相关子查询的数据库查询语句（如 SQL 语句）进行优化，在执行阶段执行优化后的数据库查询语句时，需要多次访问子查询指定的表（如采用半连接优化）或者多次访问子查询指定表对应的临时表（如采用物化优化），时间花费比较大，同时也消耗大量的计算资源，降低了具有非相关子查询的数据库查询语句的执行效率，尤其是在面临大数据量的查询时表现的更为明显。

有鉴于此，本申请实施例提供了一种非相关子查询的优化方法和装置，可以提高具有非相关子查询的数据库查询语句的执行效率。以下将分别进行详细说明。

图 1a 是本申请实施例提供的一种非相关子查询的优化方法的实施环境示意图。其中，计算设备 10 集成有本申请任一实施例提供的非相关子查询的优化装置 11，用于实现本申请任一实施例提供的非相关子查询的优化方法。该计算设备 10 与用户终端 20 之间通过网络 30 连接，所述网络 30 可以是有线网络，也可以是无网络。

本实施例将从非相关子查询的优化装置的角度进行描述，该非相关子查询的优化装置具体可以集成在计算设备中，如集成在服务器中，该服务器可以是数据库服务器等。

本申请实施例提供了一种非相关子查询的优化方法，包括：确定数据库查询语句中目标子句是否存在非相关子查询语句，若存在，则获取目标子句对应的执行结果集的预估行数，当该预估行数小于预设阈值时，执行该非相关子查询语句，得到执行结果集，根据该执行结果集对该目标子句进行改写，以消除该非相关子查询语句。

如图 1b 所示，该非相关子查询的优化方法的具体流程可以如下：

步骤 101、确定数据库查询语句中目标子句是否存在非相关子查询语句，若是，则执行步骤 102，若否，则结束非相关子查询的优化流程。

具体地，可以在数据库查询语句的逻辑优化阶段，确定数据库查询

语句中目标子句是否存在非相关子查询语句。

其中，数据库查询语句可以由多种语言实现，比如，可以由SQL（Structured Query Language，结构化查询语言）或者其他类型的语言来实现。其中，SQL可以包括DQL（Data Query Language，数据查询语言）、DML（Data Manipulation Language，数据操作语言）等语言。当数据库查询语句由SQL实现时，数据库查询语句即为SQL语句，当由DQL、DML实现时，数据库查询语句即为DQL语句、DML语句。

本实施例中，子查询是一种常用计算机语言如数据库查询语言中嵌套查询下层的程序模块。当一个查询是另一个查询的子部分时，可以称之为子查询；该子查询是嵌套在某个查询内的查询，因此，也可以称之为嵌套查询。

一般地，根据子查询中涉及的关系对象与外层关系对象间的关系，子查询可以分为相关子查询和非相关子查询两种类型。

相关子查询为：执行依赖于外层父查询的一些属性值的子查询。相关子查询因依赖于父查询的参数，当父查询的参数改变时，相关子查询需要根据新参数值重新执行（查询优化器对相关子查询进行优化有一定意义）。

非相关子查询为：不依赖于外层父查询的任何属性值的子查询，这样子查询具有独立性，可独自求解，形成一个子查询计划先于外层的查询求解。

本实施例中，非相关子查询语句一般嵌套在数据库查询语句的目标子句当中，该目标子句可以为可能出现子查询的子句，比如条件子句。以数据库查询语句为SQL语句时，目标子句可以包括：

a) 目标列位置：子查询如果位于目标列，则只能是标量子查询，否则数据库可能返回类似“错误：子查询必须只能返回一个字段”的提示。

b) FROM子句：相关子查询出现在FROM子句中，数据库可能返回类似“在FROM子句中的子查询无法参考相同查询级别中的关系”的提示，所以相关子查询不能出现在FROM子句中；非相关子查询出现在

FROM子句中，可上拉子查询到父层，在多表连接时统一考虑连接代价然后择优。

c) 条件子句如 (WHERE子句): 出现在WHERE子句中的子查询，是一个条件表达式的一部分，而表达式可以分解为操作符和操作数；根据参与运算的不同的数据类型，操作符也不尽相同，如INT型有“>、<、=、<>”等操作，这对子查询均有一定的要求（如INT型的等值操作，要求子查询必须是标量子查询）。另外，子查询出现在WHERE子句中的格式，也有用谓词指定的一些操作，如IN、BETWEEN、EXISTS等。

d) JOIN/ON子句: JOIN/ON子句可以拆分为两部分，一是JOIN块类似于FROM子句，二是ON子句块类似于WHERE子句，这两部分都可以出现子查询。子查询的处理方式同FROM子句和WHERE子句。

e) GROUPBY子句: 目标列必须和GROUPBY关联。可将子查询写在GROUPBY位置处，但子查询用在GROUPBY处没有实用意义。

f) ORDERBY子句: 可将子查询写在ORDERBY位置处。但ORDERBY操作是作用在整条SQL语句上的，子查询用在ORDERBY处没有实用意义。

步骤102、获取目标子句对应的执行结果集的预估行数。

其中，目标子句对应的执行结果集为执行该目标子句得到的结果集。该预估行数即为目标子句对应的执行计划的估计基数，可以为从数据表结果集或索引返回的估计行数。

以SQL语句为例，假设SQL语句如下（uk表示唯一键）:

```
SELECT * FROM t1
```

```
WHERE uk IN (SELECT k2_2 FROM t2 WHERE k2_2>2);
```

此时，便可以执行WHERE条件子句对应的结果集的预估行数，也即从t2表中查询到的结果集。

其中，目标子句如条件子句对应的执行结果集的预估行数可以基于表的历史估计行数和选择率得到。也即步骤“获取目标子句对应的执行结果集的预估行数”可以包括:

获取目标子句对应的待查询表的历史估计行数（记为rows）；

根据该目标子句的表达式获取相应的选择率（记为s）；

根据该历史估计行数和该选择率，获取目标子句对应的待查询表的预估行数（记为w_rows）；

将该目标子句对应的待查询表的预估行数作为目标子句对应的执行结果集的预估行数。

比如，通过公式 $w_rows = rows * s$ 便可以计算出目标子句对应的执行结果集的预估行数。

其中，待查询表的历史估计行数为此刻之前该待查询表的估计行数，一般可以在查询语句处理过程中，设置一数据统计模块来统计记录待查询表的估计行数，形成历史统计数据。

因此，本实施例中，可以从历史统计数据中获取目标子句对应的待查询表的估计行数，即历史估计行数。

例如，通过上述介绍的方式本实施例可以计算WHERE条件子句对应的执行结果集的预估行数，这个过程与常规的计算实际行数的过程一样。譬如，如果非相关子查询子句没有WHERE条件，则选择率为1，即相当于w_rows等于rows。

步骤103、当该预估行数小于预设阈值时，执行该非相关子查询语句。

在本申请一些实施例中，如果预估行数比较大时，如超过预设阈值时，可以不采用本申请实施例提供的优化方法，而采用常规优化方式。

其中，预设阈值（记为Kmax）是一个哨兵值，为正整数，表示的是本申请实施例提供的优化方法所支持的非相关子查询的结果集的大小，该预设阈值是一个系统/会话参数，其可以根据实际需求设定。另外，该预设阈值还可以动态调节，比如，该预设阈值的默认值为2000，最大值为20000（用户可定义）等。

步骤104、根据非相关子查询语句的执行结果集对该目标子句进行改写，以消除该非相关子查询语句。

具体地，当执行完非相关子查询语句时，可以根据非相关子查询语句的执行结果集对目标子句进行改写。

本实施例中根据执行结果集对目标子句改写的方式包括：直接将目标子句的表达式中的非相关子查询语句替换为执行结果以消除非相关子查询。

例如，以SQL语句为例，假设SQL语句如下（uk表示唯一键）：

```
SELECT * FROM t1
```

```
WHERE uk IN (SELECT k2_2 FROM t2 WHERE k2_2>2);
```

非相关子查询“SELECT k2_2 FROM t2 WHERE k2_2>2”被执行后，得到的执行结果为：{3,4}。

那么，可以将原先的SQL语句的WHERE子句变为：

```
WHERE uk IN {3,4}。
```

在本申请一些实施例中，还可以进一步简化目标子句如条件子句的表达式，提升数据库查询语句的执行效率。比如，在将目标子句的表达式中的非相关子查询语句替换为执行结果之后，还可以根据目标子句中的操作符对替换后的表达式进行相应的简化变形。也即步骤“根据该执行结果集对该目标子句进行改写”可以包括：

将该目标子句的表达式中的非相关子查询语句替换为非相关子查询语句的执行结果集；

根据该目标子句中的操作符，对替换后的表达式进行相应的变形优化。

例如，以SQL语句为例，假设SQL语句如下（uk表示唯一键）：

```
SELECT * FROM t1
```

```
WHERE uk IN (SELECT k2_2 FROM t2 WHERE k2_2>2);
```

非相关子查询“SELECT k2_2 FROM t2 WHERE k2_2>2”被执行后，得到的执行结果为：{3,4}。

那么，可以将原先的SQL语句的WHERE子句变为：

```
WHERE uk IN {3,4};
```

由于操作符为“IN”，因此，可以将原先的SQL语句的WHERE子句变形为：

```
WHERE uk=3 OR uk=4;
```

这样，原先的SQL语句就可以变形为：

```
SELECT * FROM t1
```

```
WHERE uk=3 OR uk=4。
```

观察上面的这条SQL语句，由原先的带有子查询的查询变为了单表扫描，且可以利用uk上的唯一索引，进行2次索引扫描就能得到查询结果，所以整个SQL语句的时间复杂度变为了O(1)，提升了该条SQL语句的执行效率。

另外，其他的非IN子查询的子句，也可以做类似变换（本文不再举例，同样适用于：<op>SOME、<op>ANY、<op>ALL、LIKE、BETWEEN...AND等操作符。也适用于：=、>、>=、<、<=等操作符。另外，也适用于IN、NOT IN、EXISTS、NOT EXISTS等操作符），比如，假设SQL语句如下（uk表示唯一键）：

```
SELECT * FROM t1
```

```
WHERE uk > SOME (SELECT k2_2 FROM t2 WHERE k2_2>2);
```

WHERE子句可以变换为：

```
uk > Min(3,4);
```

然后进一步变形为：

```
uk > 3;
```

即初始的SQL语句变形为（这个优化，适合在索引上使用范围扫描加速查询速度）：

```
SELECT * FROM t1
```

```
WHERE uk > 3。
```

为了节省资源，在本申请一些实施例中，可以在执行非相关子查询语句的过程中，如果获得的执行结果集的行数（即元组个数）大于预设阈值时停止执行非相关子查询语句，结束非相关子查询的优化流程，否

则继续执行非相关子查询语句。也即，步骤“执行该非相关子查询语句”可以包括：

执行非相关子查询语句，并在执行的过程中实时获取当前执行结果集的行数；

判断当前执行结果集的行数是否大于预设阈值；

若当前执行结果集的行数大于预设阈值，则停止执行非相关子查询语句，结束非相关子查询的优化流程；

若当前执行结果集的行数不大于预设阈值，则继续执行非相关子查询语句，返回执行步骤“在执行的过程中实时获取当前执行结果集的行数”，直到执行完非相关子查询语句为止。

此时，步骤“根据该非相关子查询语句的执行结果集对该目标子句进行改写”可以包括：当执行完非相关子查询语句时，根据该非相关子查询语句的执行结果集对该目标子句进行改写。

其中，预设阈值为前述的哨兵值Kmax，在执行非相关子查询的过程中，可以对获得的元组进行计数操作（即获取当前执行结果集的行数），如果获得的元组个数超过哨兵值Kmax，则停止子查询的执行，结束非相关子查询的优化流程。

在消除非相关子查询语句之后，本实施例提供的优化方法还可以包括：对改写后的目标子句进行相应的逻辑优化。比如，如果改写后的目标子句包含常量等式或不等式时，可以根据改写后的目标子句包含的常量等式或不等式，对改写后的目标子句的表达式进行优化。也即，步骤“对改写后的目标子句进行相应的逻辑优化”可以包括：

当改写后的目标子句包含常量等式或不等式时，根据改写后的目标子句包含的常量等式或不等式，对改写后的目标子句的表达式进行优化。

例如，以SQL语句为例，假设SQL语句如下（uk表示唯一键）：

```
SELECT * FROM t1, t3
WHERE t1.uk IN (SELECT k2_2 FROM t2 WHERE k2_2>3)
AND t1.uk=t3.uk;
```

根据非相关子查询语句“SELECT k2_2 FROM t2 WHERE k2_2>3”的执行结果可以将WHERE子句变为：

```
WHERE t1.uk=4  
      AND t1.uk=t3.uk;
```

这样，原先的SQL语句就变为：

```
SELECT * FROM t1, t3  
WHERE t1.uk=4  
      AND t1.uk=t3.uk;
```

由于经过非相关子查询优化后的WHERE子句包含常量等式，因此，可以根据常量等式对WHERE子句进行进一步逻辑优化，得到优化后的SQL语句为：

```
SELECT * FROM t1, t3  
WHERE t1.uk=4  
      AND t3.uk=4;
```

这样，对表t1和t3的连接操作，变为在这两个表上先分别作一次索引扫描，然后各自有一条元组参与连接操作，所以连接的时间复杂度变为O(1)，大大地提高了连接的效率和语句执行效率。

由上可知，本申请实施例提供的非相关子查询的优化方案可以在语句执行阶段之前如在逻辑优化阶段，预先执行非相关子查询语句，并根据执行结果来对非相关子查询语句所在的目标子句进行改写，以消除非相关子查询，这样在后续语句执行阶段中无需进行任何非相关子查询语句相关的查询和访问，节约了计算资源，节省了数据库查询语句的执行时间，提高了具有非相关子查询的数据库查询语句的执行效率。

当采用半连接优化方式对数据库查询语句进行非相关子查询的优化时，从语句执行结果来看，尽管子查询被消除，但是，连接操作存在，在执行器执行语句期间，对于子查询指定的表还是需要多次访问的，这样数据库查询语句的执行效率不会高，连接的时间花费较多。如果是大数据量的查询，多次访问表还会造成很大性能损失。

例如，以MySQL为例，说明半连接优化方式，参考图1c，从图1c中的执行结果来看，尽管子查询被消除，但是，连接操作存在。其中，id值都为1表示是同一个层次即子查询不存在了，“SHOW WARNINGS”命令的显示结果表明非相关子查询被优化为了半连接（semi join）。在执行器执行期间，对于子查询指定的表t2还是需要多次访问的，这样SQL语句的执行效率不会高，连接的时间花费为 $O(n*m)$ 。如果是大数据量的查询，多次访问表t2会耗费大量计算资源，造成很大性能损失。

然而，本申请实施例提供的非相关子查询的优化方式可以预先执行非相关子查询语句，并根据执行结果来对非相关子查询语句所在的目标子句进行改写，以消除非相关子查询，相比半连接优化方式，可以避免在语句执行阶段多次访问子查询指定的表，节省了计算资源以及语句执行时间（具体地节省了子查询和父查询做连接操作的时间），提升了语句执行效率。

当采用物化优化方式对数据库查询语句进行非相关子查询的优化时，从语句执行结果来看，子查询没有被消除，只是被物化到一个临时表中，而且连接操作存在，在执行器执行期间，对于子查询指定的表的物化后的临时表还是需要多次访问的，连接的时间花费较大，这样语句的执行效率不会高。如果是大数据量的查询，临时表需要多次访问也会耗费大量计算资源，造成很大性能损失。比如，参考图1d，以MySQL为例，说明物化优化方式，参考图1d，从图1d中的执行结果来看，查询没有被消除，只是被物化到一个临时表中（MATERIALIZED）。然后，物化的临时表<subquery2>和表t1做了半连接操作（semi join）。在执行器执行期间，对于子查询指定的表t2的物化后的临时表还是需要多次访问的，连接的时间花费为 $O(n*m)$ ，这样SQL语句的执行效率不会高。如果是大数据量的查询，临时表需要多次访问会造成很大性能损失。

然而，本申请实施例提供的非相关子查询的优化方式可以预先执行非相关子查询语句，并根据执行结果来对非相关子查询语句所在的目标子句进行改写，以消除非相关子查询，相比物化优化方式，可以避免在

语句执行阶段多次访问子查询指定表物化后的临时表，节省了计算资源以及语句执行时间（具体地节省了子查询和父查询做连接操作的时间），提升了语句执行效率。

根据上述实施例所描述的方法，以下将举例作进一步详细说明。

本申请实施例以非相关子查询的优化装置集成在计算设备，如服务器中，且数据库查询语句为SQL语句为例，来介绍本申请实施例提供的非相关子查询的优化方法。

如图2a所示，一种非相关子查询的优化方法，具体流程可以如下：

步骤201、服务器在SQL语句的逻辑优化阶段，确定SQL语句中目标子句是否存在非相关子查询语句；若是，则执行步骤202，若否，结束非相关子查询的优化流程。

其中，目标子句可以为可能出现子查询的子句，比如可以为WHERE子句、FROM子句、JOIN/ON子句等等。

步骤202、服务器从统计数据中获取目标子句对应的待查询表的历史估计行数rows。

步骤203、服务器根据历史估计行数rows和选择率s，获取目标子句对应的待查询表的预估行数w_rows。

比如，通过公式 $w_rows = rows * s$ 便可以计算出目标子句对应的执行结果集的预估行数。

例如，可以计算WHERE条件子句对应的执行结果集的预估行数，这个过程，与常规的计算实际行数的过程一样。如果非相关子查询子句没有WHERE条件，则选择率为1，即相当于w_rows等于rows。

步骤204、服务器判断预估行数是否小于预设阈值Kmax，若是，则执行步骤205，若否，则结束非相关子查询的优化流程。

其中，预设阈值Kmax表示的是本申请实施例优化方法所支持的非相关子查询的结果集的大小，该Kmax是一个系统/会话参数，其可以根据实际需求设定，另外，该Kmax还可以动态调节，比如，默认值为2000，最大值为20000（用户可定义）等。

步骤205、服务器执行非相关子查询语句，并在执行的过程中实时获取执行结果集的行数。

步骤206、服务器判断当前执行结果集的行数是否大于预设阈值Kmax，若否，则返回步骤205继续执行非相关子查询语句，若是，则停止执行非相关子查询语句，结束非相关子查询的优化流程。

比如，在执行非相关子查询语句的过程中，对获得的元组进行计数操作，如果获得的元组个数超过哨兵值Kmax，则停止子查询的执行，结束流程。

常规的优化方式不会在优化阶段调用执行器来完成查询结果的获取，本申请实施例则提出在优化阶段即可以调用执行器，已提前获得一个非相关子查询的执行结果。

其中， $w_rows > Kmax$ 意味着非相关子查询的元组个数超过哨兵值Kmax(即非相关子查询是大表)，则不再采用本申请实施例的优化方法，退出本优化方法，可以采用常规的优化方式。

步骤207、当执行完非相关子查询语句时，服务器根据当前执行结果集对该目标子句进行改写，以消除该非相关子查询语句。

比如，可以直接将目标子句的表达式中非相关子查询语句替换为执行结果以消除非相关子查询。

又比如，在将目标子句的表达式中非相关子查询语句替换为执行结果之后，还可以根据目标子句中的操作符对替换后的表达式进行相应的简化变形。具体地，将该目标子句的表达式中非相关子查询语句替换为该执行结果集；根据该目标子句中的操作符，对替换后的表达式进行相应的变形优化。

例如SQL语句如下：

```
SELECT * FROM t1
```

```
WHERE uk > SOME (SELECT k2_2 FROM t2 WHERE k2_2 > 2);
```

其中的WHERE语句可以变换为：

```
uk > Min(5,6);
```

然后进一步变形为:

$uk > 5$;

即初始的SQL语句可变形为(这个优化, 适合在索引上使用范围扫描加速查询速度):

SELECT * FROM t1

WHERE $uk > 5$ 。

在本申请一些实施例中, 在消除非相关子查询语句之后, 可以对经过非相关子查询优化后的SQL语句进行相应的逻辑优化。比如, 当非相关子查询被变为常量等式或不等式, 可以进一步优化其他表达式。非相关子查询在优化阶段执行后, 即可用于化简WHERE子句中的表达式, 这一点, 为逻辑优化提供了进一步优化的可能性。

例如查询语句如下:

SELECT * FROM t1, t3

WHERE t1.uk IN (SELECT k2_2 FROM t2 WHERE $k2_2 > 3$)

AND t1.uk=t3.uk;

结果非相关子查询的优化, 上述语句可以变形为

SELECT * FROM t1, t3

WHERE t1.uk=5

AND t1.uk=t3.uk;

然后, 在后续的逻辑优化中, 可以进一步变形为:

SELECT * FROM t1, t3

WHERE t1.uk=5

AND t3.uk=5。

从上述语句可以看出: 对表t1和t3的连接操作, 变为在这两个表上先分别作一次索引扫描, 然后各自有一条元组参与连接操作, 所以连接的时间复杂度变为 $O(1)$, 大大地提高了连接的效率。

本实施例提供的非相关子查询的优化方式可以在逻辑优化阶段实施, 参考图2b并结合图2a, 在对语句进行语法分析之后, 可以进入语句

的逻辑优化阶段，在逻辑优化阶段中采用本申请实施例提供的优化方式进行非相关子查询的优化，在进行非相关子查询的优化之后（比如退出非相关子查询的优化即结束非相关子查询的优化流程之后，或者消除非相关子查询之后），继续对语句做相应的逻辑优化，在逻辑优化完成后，可以对语句进行物理优化，在物理优化完成后，可以调用执行器执行优化后的语句，最后返回结果集。

由上可知，在本申请实施例中，确定数据库查询语句中目标子句是否存在非相关子查询语句，若存在，则获取目标子句对应的执行结果集的预估行数，当该预估行数小于预设阈值时，执行该非相关子查询语句，得到执行结果集，根据该执行结果集对该目标子句进行改写，以消除该非相关子查询语句；该方案可以在语句执行阶段之前如在逻辑优化阶段，预先执行非相关子查询语句，并根据执行结果来对非相关子查询语句所在的目标子句进行改写，以消除非相关子查询，这样在后续语句执行阶段中无需进行任何非相关子查询语句相关的查询和访问，节省了计算资源以及数据库查询语句的执行时间，提高了具有非相关子查询的数据库查询语句的执行效率。

此外，当执行结果作为SQL语句中的常量子句时，可以参与到其他类型的逻辑优化子阶段，能为SQL语句开展其他类型的优化带来更多优化的可能，进一步提升语句执行效率。比如，执行结果作为DQL/DML中的子句常量，可以参与到WHERE/JOIN-ON子句的优化过程中，即进一步为WHERE子句的优化提供了新的优化可能。

为了更好地实施以上方法，本申请实施例还提供了非相关子查询的优化装置，如图3a所示，该非相关子查询的优化装置包括：确定单元301、获取单元302、执行单元303和改写单元304。

确定单元301，用于确定数据库查询语句中目标子句是否存在非相关子查询语句。

具体地，确定单元301可以在数据库查询语句的逻辑优化阶段，确定数据库查询语句中目标子句是否存在非相关子查询语句。

其中，数据库查询语句可以由多种语言实现，比如，可以由SQL或者其他类型的语言来实现。其中，SQL可以包括DQL、DML等语言。

本实施例中，子查询是一种常用计算机语言如数据库查询语言中嵌套查询下层的程序模块。当一个查询是另一个查询的子部分时，可以称之为子查询；该子查询是嵌套在某个查询内的查询，因此，也可以称之为嵌套查询。

获取单元302，用于当确定单元301确定存在非相关子查询语句时，获取目标子句对应的执行结果集的预估行数。

其中，目标子句对应的执行结果集为执行该目标子句得到的结果集。该预估行数即为目标子句对应的执行计划的估计基数，可以为从数据表结果集或索引返回的估计行数。

本实施例中，目标子句如条件子句对应的执行结果集的预估行数可以基于表的历史估计行数和选择率得。比如，获取单元302，可以具体用于：获取目标子句对应的待查询表的历史估计行数；根据该目标子句的表达式获取相应的选择率；根据该历史估计行数和该选择率，获取目标子句对应的待查询表的预估行数；将该目标子句对应的待查询表的预估行数作为目标子句对应的执行结果集的预估行数。

比如，通过公式 $w_rows = rows * s$ 便可以计算出目标子句对应的执行结果集的预估行数。

执行单元303，用于当该预估行数小于预设阈值时，执行该非相关子查询语句。

其中，预设阈值（Kmax）是一个哨兵值，为正整数，表示的是本申请实施例提供的优化方式所支持的非相关子查询的结果集的大小，该预设阈值是一个系统/会话参数，其可以根据实际需求设定。另外，该预设阈值还可以动态调节，比如，默认值为2000，最大值为20000（用户可定义）等。

为了节省资源，在本申请一些实施例中，可以在执行非相关子查询语句的过程中，如果获得的执行结果集的行数（即元组个数）大于预设阈值时停止执行非相关子查询语句，结束非相关子查询的优化流程，否则继续执行非相关子查询语句。也即，执行单元303可以具体用于：

执行非相关子查询语句，并在执行的过程中实时获取当前执行结果集的行数；

判断当前执行结果集的行数是否大于该预设阈值；

若当前执行结果集的行数不大于预设阈值，则继续执行该非相关子查询语句，并返回实时获取当前执行结果集的行数的步骤，直到执行完非相关子查询语句为止。

若当前执行结果集的行数大于预设阈值，则停止执行非相关子查询语句，结束非相关子查询的优化。

改写单元304，用于根据该非相关子查询语句的执行结果集对该目标子句进行改写，以消除该非相关子查询语句。

比如，改写单元304，可以用于直接将目标子句的表达式中的非相关子查询语句替换为执行结果以消除非相关子查询。

又比如，在将目标子句的表达式中非相关子查询语句替换为执行结果之后，还可以根据目标子句中的操作符对替换后的表达式进行相应的简化变形，达到进一步优化目的。参考图3b，该改写单元304可以包括：

替换子单元3041，用于将该目标子句的表达式中的非相关子查询语句替换为非相关子查询语句的执行结果集；

改写子单元3042，用于根据该目标子句中的操作符，对替换后的表达式进行相应的变形优化。

参考图3c，本实施例的优化装置还可以包括：逻辑优化单元305；

逻辑优化单元305可以用于在改写单元304消除该非相关子查询语句之后对改写后的目标子句进行相应的逻辑优化。

比如，逻辑优化单元305，可以用于在改写单元304消除该非相关子查询语句之后，当改写后的目标子句包含常量等式或不等式时，根据改

写后的目标子句包含的常量等式或不等式，对改写后的目标子句的表达式进行优化。

具体实施时，以上各个单元可以作为独立的实体来实现，也可以进行任意组合，作为同一或若干个实体来实现，以上各个单元的具体实施可参见前面的方法实施例，在此不再赘述。

该非相关子查询的优化装置具体可以集成在计算设备如服务器中，该服务器可以为数据库服务器。

由上可知，在本申请实施例中，优化装置通过确定单元301确定数据库查询语句中目标子句是否存在非相关子查询语句，若存在，则由获取单元302获取目标子句对应的执行结果集的预估行数，当该预估行数小于预设阈值时，由执行单元303执行该非相关子查询语句，得到执行结果集，由改写单元304根据该执行结果集对该目标子句进行改写，以消除该非相关子查询语句；该方案可以在语句执行阶段之前如在逻辑优化阶段，预先执行非相关子查询语句，并根据执行结果来对非相关子查询语句所在的目标子句进行改写，以消除非相关子查询，这样在后续语句执行阶段中无需进行任何非相关子查询语句相关的查询和访问，节省了计算资源以及数据库查询语句的执行时间，提高了具有非相关子查询的数据库查询语句的执行效率。

相应的，本申请实施例还提供了一种计算设备，如服务器，如图4所示，其示出了本申请实施例所涉及的服务器的结构示意图，具体来讲：

该服务器400可以包括一个或者一个以上处理核心的处理器401、一个或一个以上计算机可读存储介质的存储器402、射频(Radio Frequency, RF)电路403、电源404、输入单元405、以及显示单元406等部件。本领域技术人员可以理解，图4中示出的服务器结构并不构成对服务器的限定，可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件布置。其中：

处理器401是该服务器的控制中心，利用各种接口和线路连接整个服务器的各个部分，通过运行或执行存储在存储器402内的软件程序和/

或模块，以及调用存储在存储器402内的数据，执行服务器的各种功能和处理数据，从而对服务器进行整体监控。在本申请一些实施例中，处理器401可包括一个或多个处理核心；处理器401可集成应用处理器和调制解调处理器，其中，应用处理器主要处理操作系统、用户界面和应用程序等，调制解调处理器主要处理无线通信。可以理解的是，上述调制解调处理器也可以不集成到处理器401中。

存储器402可用于存储软件程序以及模块，即机器可读指令。处理器401通过运行存储在存储器402的软件程序以及模块，从而执行各种功能应用以及数据处理。例如，处理器401可运行存储在存储器402中的非相关子查询的优化程序以实现前述的非相关子查询的优化方法。

RF电路403可用于收发信息过程中，信号的接收和发送，特别地，将基站的下行信息接收后，交由一个或者一个以上处理器401处理。另外，将涉及上行的数据发送给基站。

服务器还包括给各个部件供电的电源404（比如电池）。在本申请一些实施例中，电源可以通过电源管理系统与处理器401逻辑相连，从而通过电源管理系统实现管理充电、放电、以及功耗管理等功能。电源404还可以包括一个或一个以上的直流或交流电源、再充电系统、电源故障检测电路、电源转换器或者逆变器、电源状态指示器等任意组件。

该服务器还可包括输入单元405，该输入单元405可用于接收输入的数字或字符信息，以及产生与用户设置以及功能控制有关的键盘、鼠标、操作杆、光学或者轨迹球信号输入。

该服务器还可包括显示单元406，该显示单元406可用于显示由用户输入的信息或提供给用户的信息以及服务器的各种图形用户接口，这些图形用户接口可以由图形、文本、图标、视频和其任意组合来构成。显示单元406可包括显示面板，可选的，可以采用液晶显示器（LCD, Liquid Crystal Display）、有机发光二极管（OLED, Organic Light-Emitting Diode）等形式来配置显示面板。

具体在本实施例中，服务器中的处理器401会按照如下的指令，将

一个或一个以上的应用程序的进程对应的可执行文件加载到存储器402中，并由处理器401来运行存储在存储器402中的应用程序，从而实现各种功能，如下：

确定数据库查询语句中目标子句是否存在非相关子查询语句，若存在，则获取目标子句对应的执行结果集的预估行数，当所述预估行数小于预设阈值时，执行所述非相关子查询语句，根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，以消除所述非相关子查询语句。

其中，根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，包括：

将所述目标子句的表达式中的非相关子查询语句替换为非相关子查询语句的执行结果集；

根据所述目标子句中的操作符，对替换后的表达式进行相应的变形优化。

其中，获取目标子句对应的执行结果集的预估行数，包括：

获取目标子句对应的待查询表的历史估计行数；

根据所述目标子句的表达式获取相应的选择率；

根据所述历史估计行数和所述选择率，获取目标子句对应的待查询表的预估行数；

将所述目标子句对应的待查询表的预估行数作为所述目标子句对应的执行结果集的预估行数。

具体的操作步骤或过程，可以参考前面非相关子查询的优化方法实施例的详细描述，在此不再赘述。

本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通过程序来指令相关的硬件来完成的，该程序可以存储于一非易失性计算机可读存储介质中，该存储介质可以包括：只读存储器（ROM, Read Only Memory）、随机存取记忆体（RAM, Random Access Memory）、磁盘或光盘等。

以上对本申请实施例所提供的一种非相关子查询的优化方法和装置进行了详细介绍，本文中应用了具体个例对本申请的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本申请的方法及其核心思想；同时，对于本领域的技术人员，依据本申请的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本申请的限制。

权利要求书

1、一种非相关子查询的优化方法，应用于计算设备，包括：
确定数据库查询语句中目标子句是否存在非相关子查询语句；
若存在，则获取所述目标子句对应的执行结果集的预估行数；
当所述预估行数小于预设阈值时，执行所述非相关子查询语句；
根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，以消除所述非相关子查询语句。

2、如权利要求1所述的优化方法，所述根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，包括：

将所述目标子句的表达式中的非相关子查询语句替换为非相关子查询语句的执行结果集；

根据所述目标子句中的操作符，对替换后的表达式进行相应的变形优化。

3、如权利要求1所述的优化方法，所述获取所述目标子句对应的执行结果集的预估行数，包括：

获取所述目标子句对应的待查询表的历史估计行数；

根据所述目标子句的表达式获取相应的选择率；

根据所述历史估计行数和所述选择率，获取所述目标子句对应的待查询表的预估行数；

将所述目标子句对应的待查询表的预估行数作为所述目标子句对应的执行结果集的预估行数。

4、如权利要求1所述的优化方法，所述执行所述非相关子查询语句，包括：

执行所述非相关子查询语句，并在执行的过程中实时获取当前执行结果集的行数；

判断当前执行结果集的行数是否大于所述预设阈值；

若否，则继续执行所述非相关子查询语句，并返回实时获取当前执行结果集的行数的步骤，直到执行完所述非相关子查询语句为止。

5、如权利要求1所述的优化方法，在根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，以消除所述非相关子查询语句之后，所述优化方法还包括：

当改写后的目标子句包含常量等式或不等式时，根据改写后的目标子句包含的常量等式或不等式，对改写后的目标子句的表达式进行优化。

6、一种非相关子查询的优化装置，包括：

处理器以及与所述处理器相连接的存储器，所述存储器中存储有可由所述处理器执行的机器可读指令，所述处理器执行所述机器可读指令完成以下操作：

确定数据库查询语句中目标子句是否存在非相关子查询语句；

当确定存在非相关子查询语句时，获取目标子句对应的执行结果集的预估行数；

当所述预估行数小于预设阈值时，执行所述非相关子查询语句；

根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，以消除所述非相关子查询语句。

7、如权利要求6所述的优化装置，所述处理器执行所述机器可读指令完成以下操作：

将所述目标子句的表达式中的非相关子查询语句替换为非相关子查询语句的执行结果集；

根据所述目标子句中的操作符，对替换后的表达式进行相应的变形优化。

8、如权利要求6所述的优化装置，所述处理器执行所述机器可读指令完成以下操作：

获取所述目标子句对应的待查询表的历史估计行数；

根据所述目标子句的表达式获取相应的选择率；

根据所述历史估计行数和所述选择率，获取所述目标子句对应的待查询表的预估行数；

将所述目标子句对应的待查询表的预估行数作为所述目标子句对

应的执行结果集的预估行数。

9、如权利要求6所述的优化装置，所述处理器执行所述机器可读指令完成以下操作：

执行所述非相关子查询语句，并在执行的过程中实时获取当前执行结果集的行数；

判断当前执行结果集的行数是否大于所述预设阈值；

若否，则继续执行所述非相关子查询语句，并返回实时获取当前执行结果集的行数的步骤，直到执行完所述非相关子查询语句为止。

10、如权利要求6所述的优化装置，所述处理器执行所述机器可读指令完成以下操作；

当改写后的目标子句包含常量等式或不等式时，根据改写后的目标子句包含的常量等式或不等式，对改写后的目标子句的表达式进行优化。

11、一种非易失性计算机可读存储介质，所述存储介质中存储有机器可读指令，所述机器可读指令可以由处理器执行以完成以下操作：

确定数据库查询语句中目标子句是否存在非相关子查询语句；

若存在，则获取所述目标子句对应的执行结果集的预估行数；

当所述预估行数小于预设阈值时，执行所述非相关子查询语句；

根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，以消除所述非相关子查询语句。

12、如权利要求11所述的非易失性计算机可读存储介质，所述根据所述非相关子查询语句的执行结果集对所述目标子句进行改写，包括：

将所述目标子句的表达式中的非相关子查询语句替换为非相关子查询语句的执行结果集；

根据所述目标子句中的操作符，对替换后的表达式进行相应的变形优化。

13、如权利要求11所述的非易失性计算机可读存储介质，所述获取所述目标子句对应的执行结果集的预估行数，包括：

获取所述目标子句对应的待查询表的历史估计行数；

根据所述目标子句的表达式获取相应的选择率；

根据所述历史估计行数 and 所述选择率，获取所述目标子句对应的待查询表的预估行数；

将所述目标子句对应的待查询表的预估行数作为所述目标子句对应的执行结果集的预估行数。

14、如权利要求11所述的非易失性计算机可读存储介质，所述执行所述非相关子查询语句，包括：

执行所述非相关子查询语句，并在执行的过程中实时获取当前执行结果集的行数；

判断当前执行结果集的行数是否大于所述预设阈值；

若否，则继续执行所述非相关子查询语句，并返回实时获取当前执行结果集的行数的步骤，直到执行完所述非相关子查询语句为止。

15、如权利要求11所述的非易失性计算机可读存储介质，所述机器可读指令可以由处理器执行以完成以下操作：

当改写后的目标子句包含常量等式或不等式时，根据改写后的目标子句包含的常量等式或不等式，对改写后的目标子句的表达式进行优化。

1/7

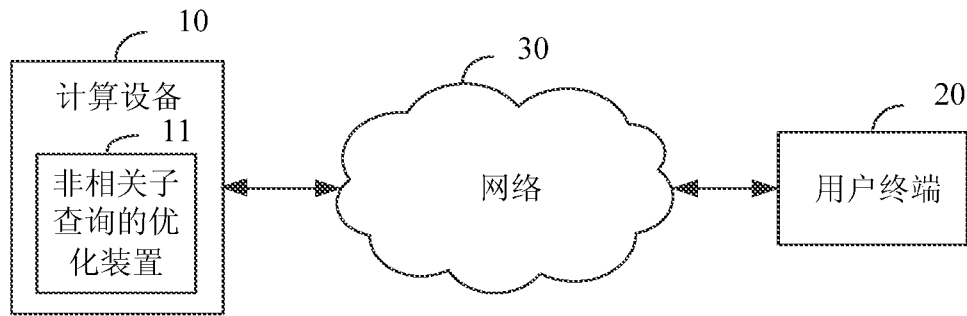


图 1a

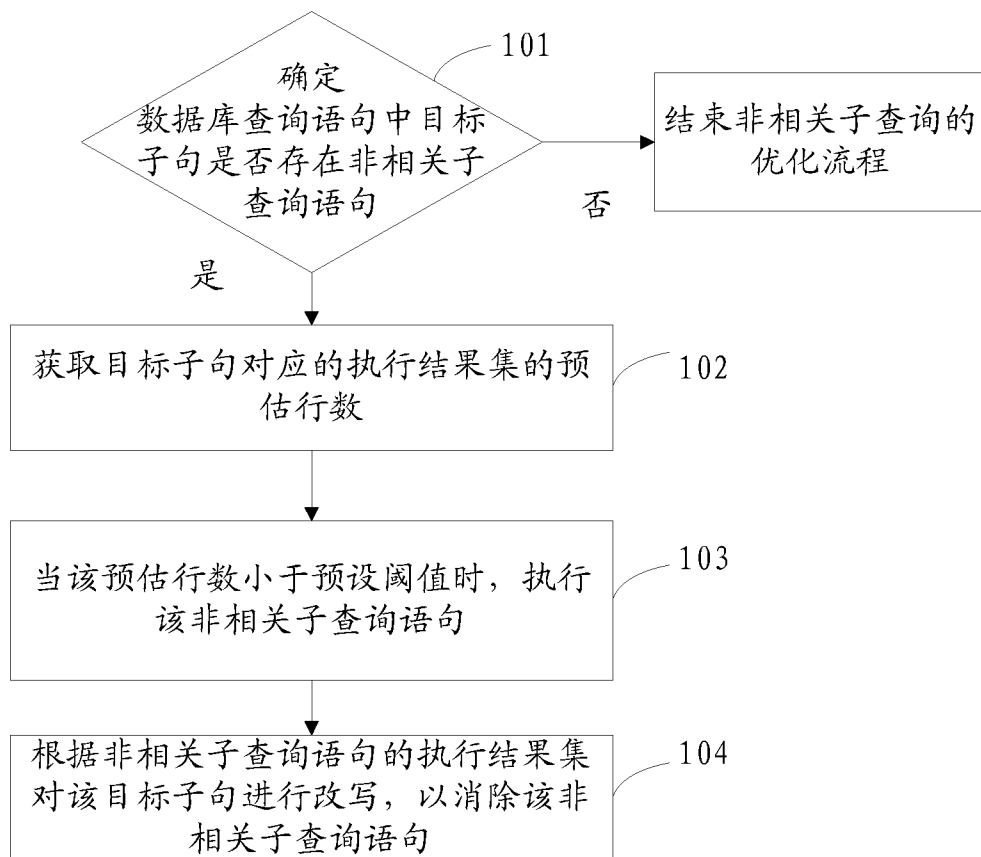


图 1b

2/7

```
mysql> EXPLAIN EXTENDED
```

```
--> SELECT ID FROM t1
```

```
--> WHERE uk IN (SELECT k2_2 FROM t2 WHERE k2_2>2);
```

+-----+-----+-----+-----+			
id	select_type	table	Extra
+-----+-----+-----+-----+			
1	SIMPLE	t1	Using where
1	SIMPLE	t2	Using where; Using index; FirstMatch(t1); Using join buffer (Block Nested Loop)
+-----+-----+-----+-----+			

```
mysql> SHOW WARNINGS;
```

```
...
```

```
/* select#1 */
```

```
select `d`.`t1`.`ID` AS `ID`
```

```
from `d`.`t1` semi join (`d`.`t2`)
```

```
where ((`d`.`t2`.`k2_2` = `d`.`t1`.`uk`) and (`d`.`t1`.`uk` > 2))
```

图 1c

3/7

```
mysql> EXPLAIN EXTENDED;
```

```
  -> SELECT ID FROM t1;
```

```
  -> WHERE col IN (SELECT col FROM t2);
```

+-----+-----+-----+-----+-----+				
id	select_type	table	Extra	>
+-----+-----+-----+-----+-----+				
_1	SIMPLE	<subquery2>	NULL	>
_1	SIMPLE	t1	Using where; Using join buffer (Block Nested Loop)	
_2	MATERIALIZED	t2	NULL	>
+-----+-----+-----+-----+-----+				

```
3 rows in set, 1 warning (0.01 sec);
```

```
^
```

```
mysql> SHOW WARNINGS;
```

```
...^
```

```
/* select#1 */ ^
```

```
select `d`.`t1`.`ID` AS `ID` ^
```

```
from `d`.`t1` semi join (`d`.`t2`) ^
```

```
where (`d`.`t1`.`col` = `<subquery2>`.`col`);
```

图 1d

4/7

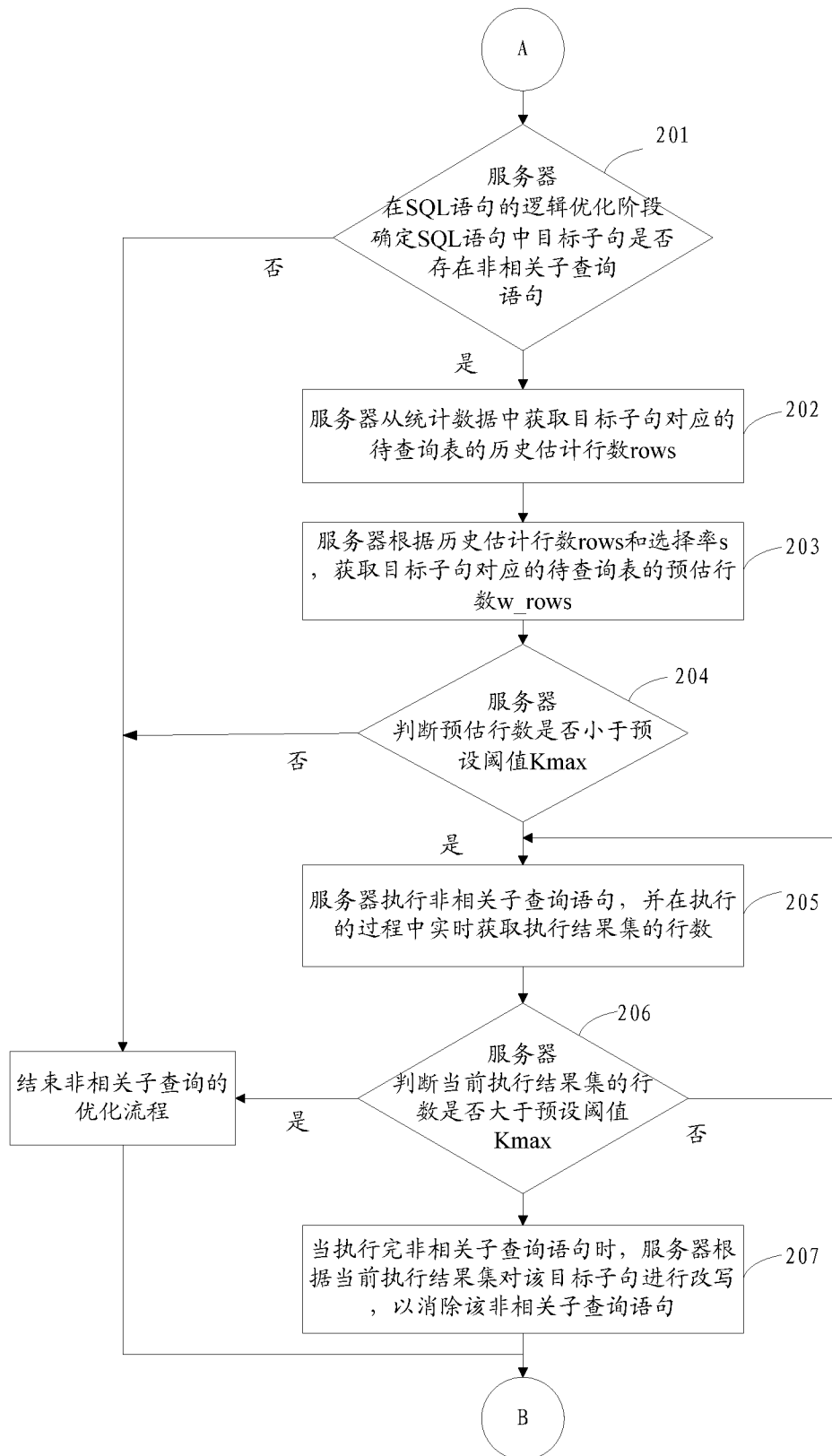


图 2a

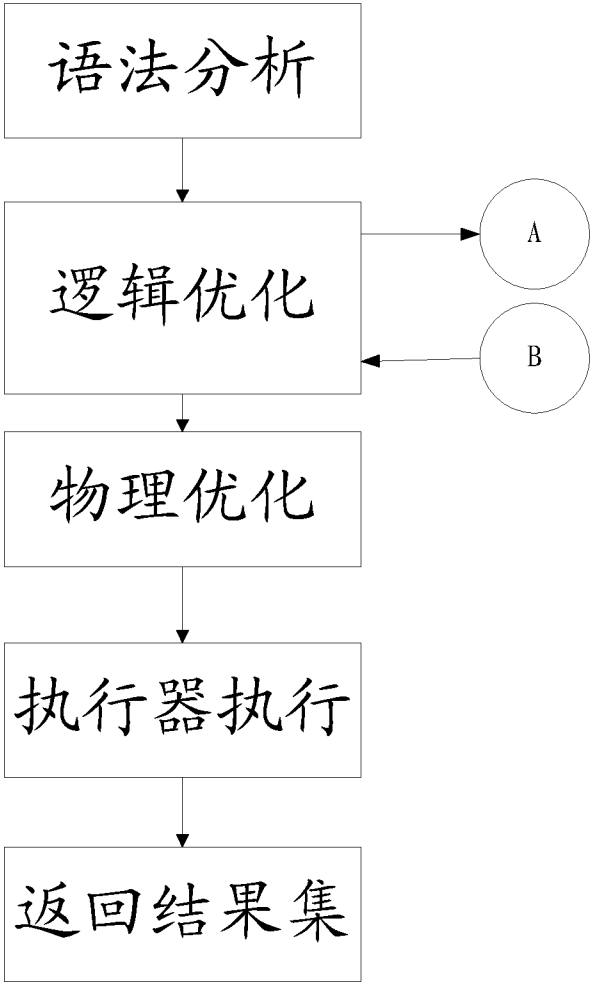


图 2b

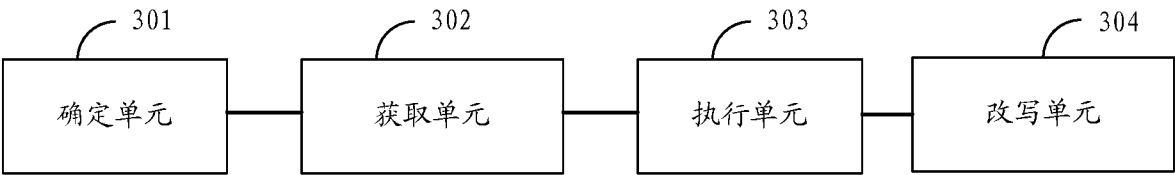


图 3a

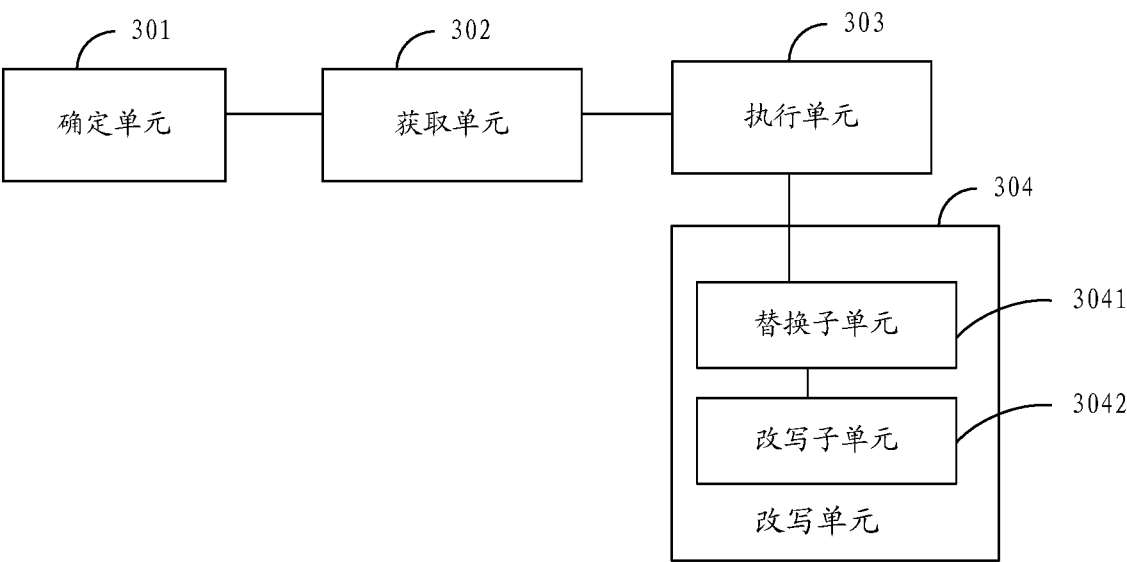


图 3b

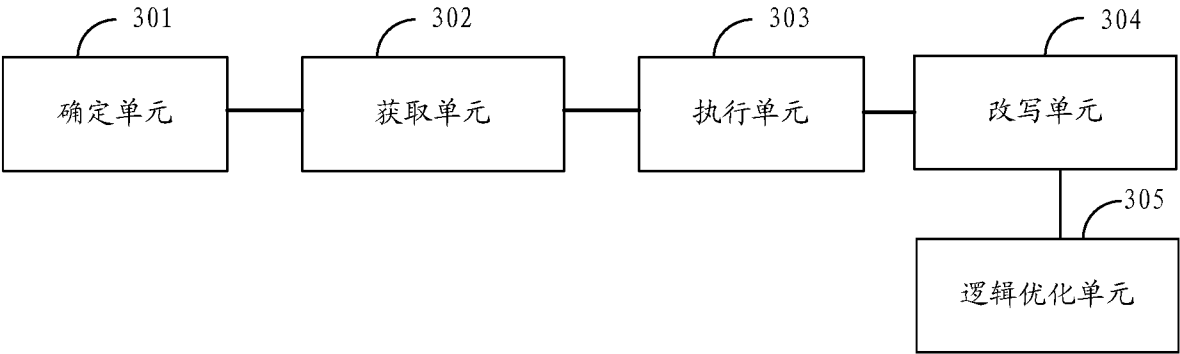


图 3c

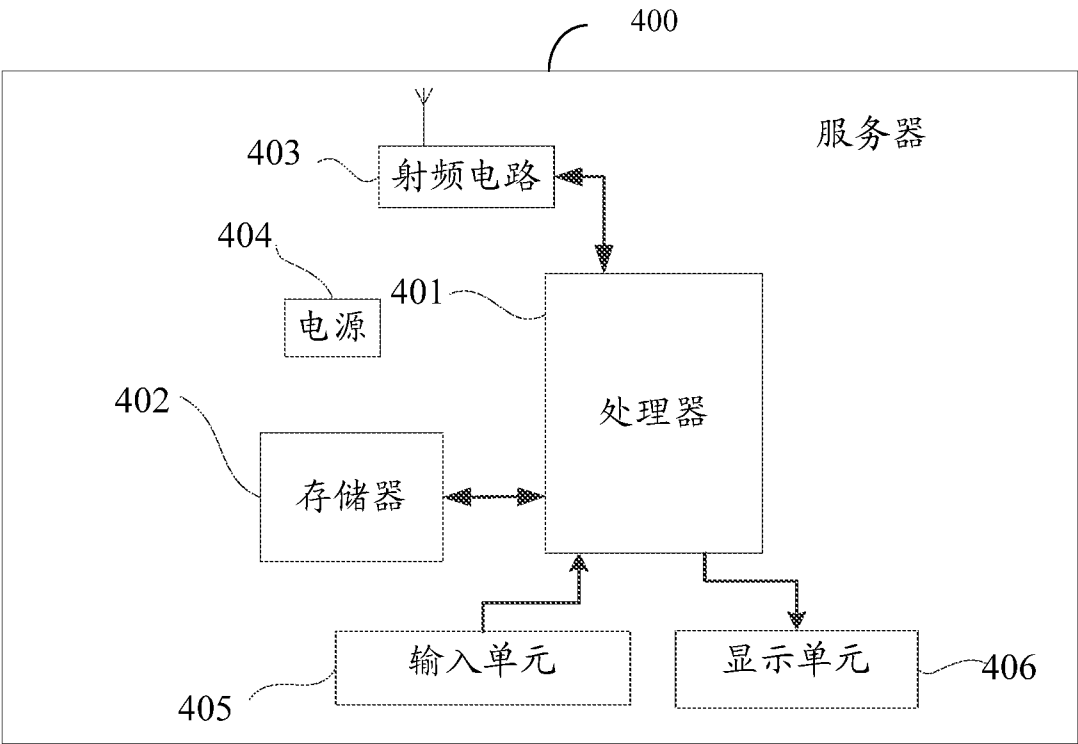


图 4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/090610

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNTXT; CNABS; VEN; USTXT; EPTXT; WOTXT; CNKI: 查询, 子查询, 消除, 优化, 重写, 改写, 预设值, 预估值, 估计值, 阈值, 结果集, 替换, 代替, query, subquery, eliminate, optimize, rewrite, preset value, predetermined value, threshold, result set, replace, instead

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 107315790 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 03 November 2017 (2017-11-03) entire document	1-15
A	CN 104765731 A (INTERNATIONAL BUSINESS MACHINES CORPORATION) 08 July 2015 (2015-07-08) entire document	1-15
A	US 2014379690 A1 (ORACLE INT CORP.) 25 December 2014 (2014-12-25) entire document	1-15
A	US 9183254 B1 (COLE, R.L. ET AL.) 10 November 2015 (2015-11-10) entire document	1-15
A	CN 103678621 A (SHANGHAI DAMENG DATABASE CO., LTD.) 26 March 2014 (2014-03-26) entire document	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

30 August 2018

Date of mailing of the international search report

10 September 2018

Name and mailing address of the ISA/CN

State Intellectual Property Office of the P. R. China
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/090610

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	107315790	A	03 November 2017	None			
CN	104765731	A	08 July 2015	CN	104765731	B	18 May 2018
US	2014379690	A1	25 December 2014	None			
US	9183254	B1	10 November 2015	US	9582539	B1	28 February 2017
				US	9665619	B1	30 May 2017
CN	103678621	A	26 March 2014	CN	103678621	B	17 August 2016

国际检索报告

国际申请号

PCT/CN2018/090610

A. 主题的分类

G06F 17/30(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNTXT;CNABS;VEN;USTXT;EPTXT;WOTXT;CNKI:查询, 子查询, 消除, 优化, 重写, 改写, 预设值, 预估值, 估计值, 阈值, 结果集, 替换, 代替, query, subquery, eliminate, optimize, rewrite, preset value, predetermined value, threshold, result set, replace, instead

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
PX	CN 107315790 A (腾讯科技深圳有限公司) 2017年 11月 3日 (2017 - 11 - 03) 全文	1-15
A	CN 104765731 A (国际商业机器公司) 2015年 7月 8日 (2015 - 07 - 08) 全文	1-15
A	US 2014379690 A1 (ORACLE INT CORP) 2014年 12月 25日 (2014 - 12 - 25) 全文	1-15
A	US 9183254 B1 (COLE, RICHARD L. 等) 2015年 11月 10日 (2015 - 11 - 10) 全文	1-15
A	CN 103678621 A (上海达梦数据库有限公司) 2014年 3月 26日 (2014 - 03 - 26) 全文	1-15

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2018年 8月 30日

国际检索报告邮寄日期

2018年 9月 10日

ISA/CN的名称和邮寄地址

中华人民共和国国家知识产权局(ISA/CN)
中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

受权官员

梁静静

电话号码 86-(20)-28958273

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2018/090610

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	107315790	A	2017年 11月 3日	无			
CN	104765731	A	2015年 7月 8日	CN	104765731	B	2018年 5月 18日
US	2014379690	A1	2014年 12月 25日	无			
US	9183254	B1	2015年 11月 10日	US	9582539	B1	2017年 2月 28日
				US	9665619	B1	2017年 5月 30日
CN	103678621	A	2014年 3月 26日	CN	103678621	B	2016年 8月 17日

表 PCT/ISA/210 (同族专利附件) (2015年1月)