[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
H04L 9/32

[12] 发明专利申请公开说明书

[21] 申请号 200380102441.2

[43] 公开日 2005 年 12 月 14 日

[11] 公开号 CN 1708945A

[22] 申请日 2003.10.17
[21] 申请号 200380102441.2
[30] 优先权
[32] 2002.10.31 [33] US [31] 10/284,944
[86] 国际申请 PCT/US2003/033135 2003.10.17
[87] 国际公布 WO2004/043000 英 2004.5.21
[85] 进入国家阶段日期 2005.4.29
[71] 申请人 国际商业机器公司
地址 美国纽约阿芒克
[72] 发明人 理查德·D·德廷杰
理查德·J·史蒂文斯

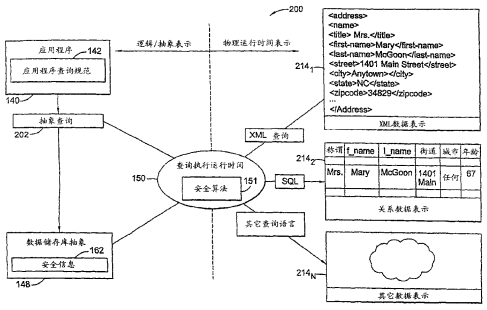
[74] 专利代理机构 北京市柳沈律师事务所
代理人 邸万奎 黄小临

权利要求书 6 页 说明书 21 页 附图 10 页

[54] 发明名称 用于可能的安全性暴露的早期告警指示的查询返回数据分析方法

[57] 摘要

用于保护数据的系统、方法和工业产品。分析查询(202)以检测安全侵害尝试。在一个实施例中，实现了用于检测选定安全侵害模式的算法(151)。通常，可以在执行查询(202)之前和执行查询(202)之后检测模式。说明性的模式包括联合查询分析(340)、削减分析(342)、不重叠(338)以及其它。



ISSN1008-4274

1. 一种对数据提供安全性的方法，包括:

接收由用户发出的对数据库的查询；以及

基于以下的至少一项来确定是否存在安全侵害模式:

(i)对所述查询相对于至少一个其它先前从该用户发出的查询进行的执行前比较分析；以及

(ii)对从所述查询的执行返回的结果和从所述至少一个其它先前发出的查询的执行返回的结果进行的执行后比较分析。

2. 如权利要求 1 所述的方法，其特征在于，所述至少一个其它先前发出的查询仅包括从用户的当前登录会话起的那些查询。

3. 如权利要求 1 所述的方法，其特征在于，基于步骤(i)确定是否存在安全侵害模式包括：确定所述查询和所述至少一个其它先前发出的查询之间的相对共同性。

4. 如权利要求 3 所述的方法，其特征在于，基于步骤(i)确定是否存在安全侵害模式还包括:

确定相对共同性是否小于预先定义的值；以及

如果是这样，则调用安全规则。

5. 如权利要求 3 所述的方法，其特征在于，基于步骤(ii)确定是否存在安全侵害模式还包括:

确定从所述查询的执行返回的结果和从所述至少一个其它先前发出的查询的执行返回的结果之间的共同结果的数目是否减小;

如果是这样，则确定相对共同性是否小于预先定义的值，并且如果是，则调用安全规则。

6. 如权利要求 1 所述的方法，其特征在于，基于步骤(i)确定是否存在安全侵害模式包括：检测所述查询和所述至少一个其它先前发出的查询的共同查询条件被配置为返回不重叠结果。

7. 如权利要求 1 所述的方法，其特征在于，基于步骤(i)确定是否存在安全侵害模式包括：检测用户获取未经授权的数据库量的企图，该企图以所述查询和所述至少一个其它先前发出的查询的、被配置为返回至少部分不重叠的结果的一个或多个共同查询条件的存在为特征。

8. 如权利要求 1 所述的方法，其特征在于，仅在所述查询和所述至少一个其它先前发出的查询被配置为访问共同表列时，才进行基于步骤(i)确定是否存在安全侵害模式。

9. 如权利要求 1 所述的方法，其特征在于，基于步骤(ii)确定是否存在安全侵害模式包括：检测从所述查询的执行返回的结果和从所述至少一个其它先前发出的查询的执行返回的结果是不重叠的。

10. 如权利要求 1 所述的方法，其特征在于，基于步骤(ii)确定是否存在安全侵害模式包括：检测结果子集构造的模式。

11. 如权利要求 1 所述的方法，还包括：如果存在安全侵害模式，则调用安全规则。

12. 如权利要求 11 所述的方法，其特征在于，当进行步骤(i)之后确定存在安全侵害模式时调用安全规则包括：终止所述查询。

13. 如权利要求 11 所述的方法，其特征在于，当进行步骤(ii)之后确定存在安全侵害模式时调用安全规则包括：阻止用户获得从所述查询的执行返回的结果。

14. 一种对数据提供安全性的方法，包括：

从用户接收多个查询；

执行对数据库的所述多个查询；

接收用户发出的对数据库的后续查询；以及

基于所述多个查询和后续查询，有计划地确定是否可以识别用户从数据库访问未经授权的数据量的尝试。

15. 如权利要求 14 所述的方法，其特征在于，有计划地确定包括：检测所述后续查询和多个查询的共同查询条件被配置为返回至少部分不重叠的结果。

16. 一种对数据提供安全性的方法，包括：

从用户接收多个查询；

执行对数据库的所述多个查询；

接收用户发出的对数据库的后续查询；

执行该后续查询；以及

基于所述多个查询和后续查询，有计划地确定是否可以识别用户绕开防止唯一地识别个人的安全限制的尝试。

17. 如权利要求 16 所述的方法，其特征在于，有计划地确定包括:

确定后续查询和所述多个查询之间的相对共同性;

确定从后续查询的执行返回的结果和从所述多个查询的执行返回的结果之间的共同结果数目是否减小;

如果是这样，则确定相对共同性是否小于预先定义的值，并且如果是，则调用安全规则。

18. 如权利要求 16 所述的方法，其特征在于，有计划地确定包括: 检测结果子集构造的模式。

19. 一种为具有特定物理数据表示的数据提供安全性的方法，包括:

提供包括用于定义抽象查询的多个逻辑字段的查询规范;

提供将所述多个逻辑字段映射到数据的物理实体的映射规则;

提供安全规则;

接收用户发出的对数据的抽象查询，其中，该抽象查询根据查询规范来定义，并配置有至少一个逻辑字段值; 以及

相对于至少一个先前从用户接收的抽象查询，分析该抽象查询，以检测促使调用安全规则的安全侵害活动的存在性。

20. 如权利要求 19 所述的方法，其特征在于，分析该抽象查询和至少一个先前从用户接收的抽象查询以检测安全侵害活动的存在性包括: 进行对该抽象查询和至少一个其它先前从用户发出的抽象查询的执行前比较分析。

21. 如权利要求 19 所述的方法，其特征在于，分析该抽象查询和至少一个先前从用户接收的抽象查询以检测安全侵害活动的存在性包括: 进行对从该抽象查询的执行返回的结果和从该至少一个其它先前发出的抽象查询的执行返回的结果的执行后比较分析。

22. 如权利要求 19 所述的方法，还包括:

检测安全侵害活动的存在性; 以及

调用安全规则。

23. 一种包含指令的计算机可读介质，当执行所述指令时，进行安全侵害识别操作，包括:

接收由用户发出的对数据库的查询; 以及

基于以下的至少一项来确定是否存在安全侵害模式:

(i)对所述查询相对于至少一个其它先前从该用户发出的查询进行的

执行前比较分析；以及

(ii)对从所述查询的执行返回的结果和从所述至少一个其它先前发出的查询的执行返回的结果进行的执行后比较分析。

24. 如权利要求 23 所述的计算机可读介质，其特征在于，所述至少一个其它先前发出的查询仅包括从用户的当前登录会话起的那些查询。

25. 如权利要求 23 所述的计算机可读介质，其特征在于，基于步骤(i)确定是否存在安全侵害模式包括：确定所述查询和所述至少一个其它先前发出的查询之间的相对共同性。

26. 如权利要求 25 所述的计算机可读介质，其特征在于，基于步骤(i)确定是否存在安全侵害模式还包括：

确定相对共同性是否小于预先定义的值；以及

如果是这样，则调用安全规则。

27. 如权利要求 25 所述的计算机可读介质，其特征在于，基于步骤(ii)确定是否存在安全侵害模式还包括：

确定从所述查询的执行返回的结果和从所述至少一个其它先前发出的查询的执行返回的结果之间的共同结果数目是否减小；

如果是这样，则确定相对共同性是否小于预先定义的值，并且如果是，则调用安全规则。

28. 如权利要求 23 所述的计算机可读介质，其特征在于，基于步骤(i)确定是否存在安全侵害模式包括：检测所述查询和所述至少一个其它先前发出的查询的共同查询条件被配置为返回不重叠结果。

29. 如权利要求 23 所述的计算机可读介质，其特征在于，基于步骤(i)确定是否存在安全侵害模式包括：检测用户获取未经授权的数据库量的企图，该企图以所述查询和所述至少一个其它先前发出的查询的、被配置为返回至少部分不重叠的结果的一个或多个共同查询条件的存在为特征。

30. 如权利要求 23 所述的计算机可读介质，其特征在于，仅在所述查询和所述至少一个其它先前发出的查询被配置为访问共同表列时，才进行基于步骤(i)确定是否存在安全侵害模式。

31. 如权利要求 23 所述的计算机可读介质，其特征在于，基于步骤(ii)确定是否存在安全侵害模式包括：检测从所述查询的执行返回的结果和从所述至少一个其它先前发出的查询的执行返回的结果是不重叠的。

32. 如权利要求 23 所述的计算机可读介质，其特征在于，基于步骤(ii)确定是否存在安全侵害模式包括：检测结果子集构造的模式。

33. 如权利要求 23 所述的计算机可读介质，还包括，如果存在安全侵害模式，则调用安全规则。

34. 如权利要求 33 所述的计算机可读介质，其特征在于，当进行步骤(i)之后确定存在安全侵害模式时调用安全规则包括：终止所述查询。

35. 如权利要求 33 所述的计算机可读介质，其特征在于，当进行步骤(ii)之后确定存在安全侵害模式时调用安全规则包括：阻止用户获得从所述查询的执行返回的结果。

36. 一种包含安全验证指令的计算机可读介质，当执行所述安全验证指令时，进行安全验证操作，包括：

从用户接收多个查询；

执行对数据库的所述多个查询；

接收用户发出的对数据库的后续查询；以及

基于所述多个查询和后续查询，有计划地确定是否可以识别用户从数据库访问未经授权的数据量的尝试。

37. 如权利要求 36 所述的计算机可读介质，其特征在于，有计划地确定包括：检测后续查询和所述多个查询的共同查询条件被配置为返回至少部分不重叠的结果。

38. 一种包含安全验证指令的计算机可读介质，当执行所述安全验证指令时，进行安全验证操作，包括：

从用户接收多个查询；

执行对数据库的所述多个查询；

接收用户发出的对数据库的后续查询；

执行该后续查询；以及

基于所述多个查询和后续查询，有计划地确定是否可以识别用户绕开防止唯一地识别个人的安全限制的尝试。

39. 如权利要求 38 所述的计算机可读介质，其特征在于，有计划地确定包括：

确定后续查询和所述多个查询之间的相对共同性；

确定从后续查询的执行返回的结果和从所述多个查询的执行返回的结果

之间的共同结果数目是否减小；

如果是这样，则确定相对共同性是否小于预先定义的值，并且如果是，则调用安全规则。

40. 如权利要求 38 所述的计算机可读介质，其特征在于，有计划地确定包括：检测结果子集构造的模式。

41. 一种包括存储在其上的信息的计算机可读介质，该信息包括：

查询规范，包括用于定义抽象查询的多个逻辑字段；

将所述多个逻辑字段映射到数据的物理实体的多个映射规则；

多个安全规则；

运行时间组件，可执行以响应于接收到用户发出的对数据的抽象查询而执行安全侵害活动检测操作，其中，该抽象查询根据查询规范来定义，并配置有至少一个逻辑字段值，所述安全侵害活动检测操作包括：

接收用户发出的对数据的抽象查询，其中，该抽象查询根据查询规范来定义，并配置有至少一个逻辑字段值；以及

相对于至少一个先前从用户接收的抽象查询，分析该抽象查询，以检测促使调用安全规则的安全侵害活动的存在性。

42. 如权利要求 41 所述的计算机可读介质，其特征在于，分析所述抽象查询和所述至少一个先前从用户接收的抽象查询以检测安全侵害活动的存在性包括：进行对所述抽象查询和至少一个其它先前从用户发出的抽象查询的执行前比较分析。

43. 如权利要求 41 所述的计算机可读介质，其特征在于，分析所述抽象查询和所述至少一个先前从用户接收的抽象查询以检测安全侵害活动的存在性包括：进行对从所述抽象查询的执行返回的结果和从所述至少一个其它先前发出的抽象查询的执行返回的结果的执行后比较分析。

44. 如权利要求 41 所述的计算机可读介质，还包括：

检测安全侵害活动的存在性；以及

调用安全规则。

45. 如权利要求 41 所述的计算机可读介质，其特征在于，安全规则防止执行所述抽象查询。

46. 如权利要求 41 所述的计算机可读介质，其特征在于，安全规则定义为记录从用户接收到抽象查询。

用于可能的安全性暴露的早期告警指示的查询返回数据分析方法

技术领域

本发明一般地涉及数据处理，并特别涉及保护数据库免受不当或未经授权的访问的方法。

背景技术

数据库是计算机化的信息存储和检索系统。关系数据库管理系统是使用用于存储和检索数据的关系技术的计算机数据库管理系统(DBMS)。最普遍的数据库类型是关系数据库，即，将数据定义为可以用多种不同方式识别和访问它的表式数据库。

无论具体架构是什么，在 DBMS 中，请求实体(例如，应用程序或操作系统)通过发出数据库访问请求而要求访问指定数据库。例如，这种请求可包括简单目录查找请求或用来在数据库中读取、改变和添加指定记录的事务或事务的组合。这些请求使用诸如结构化查询语言(SQL)的高级查询语言而做出。作为说明，SQL 用来进行交互式查询，所述交互式查询用于从诸如国际商业机器公司(IBM)的 DB2、Microsoft(微软公司)的 SQL Server、以及来自 Oracle、Sybase 和 Computer Associates 的数据库产品的数据库中获取信息，并更新数据库。术语“查询”是指一组用于从所存储的数据库中检索数据的命令。查询采用命令语言的形式，该命令语言使程序员和程序选择、插入、更新、找出数据的位置等。

在数据库的环境中，一个重要问题是安全性。数据库常常包含需要一定程度的安全性以避免被访问的机密或敏感的材料。例如，医疗记录被认为是高度私人化且机密的。因而，通常将对医疗记录的访问权限制为所选择的用户。为此，传统数据库管理系统可实施指定权限等级的用户概要(profile)。用户是否能访问某些特定的数据将取决于在他们各自的概要中指定的该用户的权限等级。

然而，前述途径是极不灵活且静态的。实际上，这种途径可能防止用户

访问比所希望的更宽的数据范围。因此，数据库的效率可能受到极大限制。另一方面，如果安全性过于放松，则敏感数据可能被泄漏。所需要的是数据可访问性和安全性的平衡。

为了说明传统数据库的缺点，例如，考虑这样的医疗数据库，其中，允许用户查看的唯一结果是门诊号码，以便保证在数据库中具有记录的病人的匿名性。用户仍然能够通过使用该用户已知的信息发出一系列精心构思的查询，来以相当程度的可靠性确定病人的身份。在这里，将这种过程称为查询联合分析。下面是设计用来根据门诊号码(该号码是唯一地识别个人的标识符)和每个查询返回的多个唯一的病人记录识别特定个人的说明性查询系列。

查询	结果
1998 年诊断患有 Alzheimer's(阿尔茨海默氏病)的人	1200
已婚并居住在 California(加利福尼亚)的人	6000
70 岁和 80 岁之间的活着的人	14000
在 1999 和 2001 年有过门诊就诊而在其它任何年份中没有门诊就诊的人	6000

独立进行时，前述查询的每一个都返回适当数目的结果。然而，总起来说，满足每个条件的结果的数目将小得多，可能只有一个人。在确定一个个人的门诊号码之后，用户可以进行返回门诊号码和任何其它信息的任何查询，并识别哪个信息对应这一个个人。

上文只是用户可如何利用传统数据库的一个示例。可以使用各种其它破坏性技术来绕开适当设置以保护数据库中包含的数据的安全机制。

因此，需要用于数据库的改进安全机制。

发明内容

本发明一般地旨在一种用于数据库安全性的方法、系统和工业产品。

在一个实施例中，提供一种对数据提供安全性的方法。一个实施例包括：用户发出的对数据库的查询；以及基于以下的至少一项来确定是否存在安全侵害模式(pattern)：(i)对所述查询相对于至少一个其它先前从用户发出的查询进行的执行前比较分析；以及(ii)对从所述查询的执行返回的结果和从至少一个其它先前发出的查询的执行返回的结果进行的执行后比较分析。

对数据提供安全性的另一方法包括：从用户接收多个查询；执行对数据库的所述多个查询；接收用户发出的对数据库的后续查询；以及基于所述多个查询和后续查询，有计划地确定是否可以识别用户从数据库访问未经授权的数据量的尝试(effort)。

对数据提供安全性的另一方法包括：从用户接收多个查询；执行对数据库的所述多个查询；接收用户发出的对数据库的后续查询；执行该后续查询；以及基于所述多个查询和后续查询，有计划地确定是否可以识别用户绕开防止唯一地识别个人的安全性限制的尝试。

另一方法提供具有特定物理数据表示(representation)的数据的安全性，该方法包括：提供包括用于定义抽象查询(abstract query)的多个逻辑字段的查询规范(specification)；提供将所述多个逻辑字段映射到数据的物理实体的映射规则；提供安全规则；接收用户发出的对数据库的抽象查询，其中，抽象查询根据查询规范来定义，并配置有至少一个逻辑字段值；以及相对于至少一个先前从用户接收的抽象查询，分析该抽象查询，以检测促使调用安全规则的安全侵害活动的存在性。

另一实施例提供一种包含指令的计算机可读介质，当执行所述指令时，进行安全侵害识别操作，包括：接收用户发出的对数据库的查询；以及基于以下的至少一项来确定是否存在安全侵害模式：(i)对所述查询相对于至少一个其它先前从用户发出的查询进行的执行前比较分析；以及(ii)对从所述查询的执行返回的结果和从所述至少一个其它先前发出的查询的执行返回的结果进行的执行后比较分析。

另一实施例提供一种包含安全验证指令的计算机可读介质，当执行所述安全验证指令时，进行安全验证操作，包括：从用户接收多个查询；执行对数据库的所述多个查询；接收用户发出的对数据库的后续查询；执行该后续查询，以及基于所述多个查询和后续查询，有计划地确定是否可以识别用户绕开防止唯一地识别个人的安全限制的尝试。

另一实施例提供一种包括存储在其上的信息的计算机可读介质，所述信息包括：查询规范，包括用于定义抽象查询的多个逻辑字段；将所述多个逻辑字段映射到数据的物理实体的多个映射规则；多个安全规则；运行时间组件，可执行以响应于接收到用户发出的对数据的抽象查询，而进行安全侵害活动检测操作，其中，抽象查询根据查询规范来定义，并配置有至少一个逻

辑字段值。安全侵害活动检测操作包括：接收用户发出的对数据的抽象查询，其中，抽象查询根据查询规范来定义，并配置有至少一个逻辑字段值；以及相对于至少一个先前从用户接收的抽象查询，分析所述抽象查询，以检测促使调用安全规则的安全侵害活动的存在性。

附图说明

因此，通过参考在附图中示出的本发明的实施例，可获得并详细地理解的本发明的上述特征，并对在上面简要概括的本发明进行更具体的描述。

然而，应当注意，附图仅示出本发明的典型实施例，并因此不能被认为是对本发明范围的限制，这是因为本发明可允许其它同等有效的实施例。

图 1 是计算机系统的一个实施例；

图 2A 是本发明一个实施例的软件组件的逻辑/物理视图；

图 2B 是抽象查询和抽象(abstraction)的数据储存库(repository)的逻辑视图；

图 3 是示出运行时间组件的操作的流程图；

图 4 是示出运行时间组件的操作的流程图；

图 5 是示出用于使用执行前分析来识别和处理不重叠条件的运行时间组件的操作的流程图；

图 6 是示出使用执行后结果分析来识别和处理不重叠条件的运行时间组件的操作的流程图；

图 7 是示出使用执行后结果分析来识别和处理查询联合分析的运行时间组件的操作的流程图；以及

图 8 是示出使用执行后结果分析来识别和处理削减(pare down)分析的运行时间组件的操作的流程图。

具体实施方式

引言

本发明一般地旨在一种用于确定用户未经授权的存取数据的企图(attempt)的系统、方法和制造产品。通常，在执行之前对查询进行分析，以及/或者对查询的执行所返回的结果进行分析。在一个实施例中，检测可能的安全侵害使得采取一种或多种安全措施。例如，在一个实施例中，不执行用户

的查询。在另一实施例中，记录该事件，并且/或者向管理员通知该事件。

在一个实施例中，安全特征被作为数据的逻辑模型的一部分而实现。逻辑模型作为数据储存库抽象层来实现，该数据储存库抽象层提供底层数据储存库的逻辑视图。以这一方式，使数据与物理地表示该数据的特定方式无关。还提供了查询抽象层，并且，其基于数据储存库抽象层。运行时间组件进行将抽象查询转换为可对特定物理数据表示使用的形式的转换。然而，尽管在此描述的抽象模型提供了本发明的一个或多个实施例，但本领域技术人员将认识到，可以在没有抽象模型的情况下实施在此提供的概念，同时仍然提供相同或相似的结果。

本发明的一个实施例作为与计算机系统，例如在图 1 中示出并在下面描述的计算机系统，一起使用的程序产品来实现。该程序产品的程序定义了实施例(包括在此描述的方法)的功能，并且可被包含在各种信号承载介质上。说明性的信号承载介质包括但不限于：(i)永久存储在不可写入存储介质(例如，诸如可由 CD-ROM 驱动器读取的 CD-ROM 盘的计算机内的只读存储设备)上的信息；(ii)存储在可写入存储介质(例如，盘驱动器中的软盘或硬盘驱动器)上的可改变信息；或(iii)通过通信介质，例如通过计算机或包括无线通信的电话网络，而传送给计算机的信息。后面的实施例具体包括从因特网和其它网络下载的信息。当这种信号承载介质承载指引本发明的功能的计算机可读指令时，其代表本发明的实施例。

通常，被执行以实现本发明实施例的例行程序可以是操作系统或特定应用程序、组件、程序、模块、对象或指令序列的一部分。本发明的软件通常包括大量指令，所述指令将被本地计算机转换为机器可读的格式，并从而产生可执行指令。此外，程序包括局部存在于该程序中或者在存储器中或者在存储设备上找到的变量和数据结构。此外，在本发明的特定实施例中，可以基于为之实现在下文中描述的各种程序的应用程序来识别所述各种程序。然而，应当认识到，以下任何特定术语都只是为了方便而使用的，因而本发明不应被限制为仅用于由这样的术语指出和/或暗含的任何特定应用程序。

环境的物理视图

图 1 示出了可实施本发明实施例的联网系统 100 的方框图。通常，联网系统 100 包括客户端(例如，用户的)计算机 102(示出了 3 台这种客户端计算机 102)、和至少一台服务器 104(一台这种服务器 104)。客户端计算机 102 和

服务器计算机 104 通过网络 126 连接。通常，网络 126 可以是局域网(LAN)和/或广域网(WAN)。在特定实施例中，网络 126 是因特网。

客户端计算机 102 包括通过总线 130 连接到存储器 112、存储设备 114、输入设备 116、输出设备 119 以及网络接口设备 118 的中央处理单元(CPU)110。输入设备 116 可以是为客户端计算机 102 提供输入的任何设备。例如，可以使用键盘、小键盘、光笔、触摸屏、轨迹球、或语音识别单元、音频/视频播放器等。输出设备 119 可以是为用户提供输出的任何设备，例如，任何传统显示屏。尽管输出设备 119 被示出为与输入设备 116 分离，但是输出设备 119 和输入设备 116 可以组合在一起。例如，可以使用具有集成触摸屏的显示屏、具有集成键盘的显示器、或组合了文本语音转换器的语音识别单元。

网络接口设备 118 可以是被配置为允许客户端计算机 102 和服务器计算机 104 之间通过网络 126 的网络通信的任何登录/退出设备。例如，网络接口设备 118 可以是网络适配器或其它网络接口卡(NIC)。

存储设备 114 优选是直接存取存储器设备(DASD)。尽管它被示出为单个单元，但它可以是固定和/或可拆卸存储设备的组合，例如固定的盘驱动器、软盘驱动器、磁带驱动器、可拆卸存储卡、或光学存储器。存储器 112 和存储设备 114 可以是跨越多个主和辅存储设备的一个虚拟地址空间的一部分。

存储器 112 优选是足够大以保存本发明的必要程序和数据结构的随机存取存储器。尽管存储器 112 被示出为单个实体，但应当理解，存储器 112 实际上可包括多个模块，并且存储器 112 可以以从高速寄存器和高速缓冲存储器到速度较低但较大的 DRAM 芯片的多个等级存在。

作为说明，存储器 112 包含操作系统 124。可有利地使用的说明性操作系统包括 Linux 和 Microsoft 的 Windows®。更一般地，可以使用支持在此公开的功能的任何操作系统。

存储器 112 还被示出为包含浏览器程序 122，当在 CPU 110 上执行浏览器程序 122 时，浏览器程序 122 提供对于在各种服务器 104 之间遨游、并将网络地址定位于一个或多个服务器 104 上的支持。在一个实施例中，浏览器程序 122 包括基于网络的图形用户接口(GUI)，其允许用户显示超文本标记语言(HTML)信息。然而，更一般地，浏览器程序 122 可以是能够呈现从服务器计算机 104 传送的信息的任何程序(优选地是基于 GUI)。

可以按照类似于客户端计算机 102 的方式而物理地布置服务器计算机

104。因此，服务器计算机 104 被一般地示出为包括通过总线 136 而互相耦接的 CPU 130、存储器 132 和存储设备 134。存储器 132 可以是足够大以保存位于服务器计算机 104 上的必要程序和数据结构的随机存取存储器。

服务器计算机 104 通常受被示出为驻留在存储器 132 中的操作系统 138 的控制。操作系统 138 的示例包括 IBM OS/400®、UNIX、Microsoft Windows® 等。更一般地，可以使用能够支持在此描述的功能的任何操作系统。

存储器 132 还包括一个或多个应用程序 140 和抽象查询接口 146。应用程序 140 和抽象查询接口 146 是包括多个指令的软件产品，所述指令在各种时刻驻留在计算机系统 100 内的各种存储器和存储设备中。当被服务器 104 中的一个或多个处理器 130 读取并执行时，应用程序 140 和抽象查询接口 146 使计算机系统 100 进行执行实施本发明各种方面的步骤或元素(element)所必需的步骤。应用程序 140(以及更一般地，包括操作系统 138 以及处于最高等级上的用户的任何请求实体)发出对数据库(例如，统称为数据库 156 的数据库 $156_1 \ldots 156_N$)的查询。作为说明，数据库 156 被示出为存储设备 134 中的数据库管理系统(DBMS)的一部分。数据库 156 代表与特定物理表示无关的任何数据集合。作为说明，可以根据关系大纲(schema) (可通过 SQL 查询访问)或根据 XML 大纲(可通过 XML 查询访问)来组织数据库 156。然而，本发明不限于特定大纲，并且考虑了向当前未知的大纲的延伸。如在此使用的，术语“大纲”一般地指数据的特定排列。

在一个实施例中，根据每个应用程序 140 包括的应用程序查询规范 142 来定义应用程序 140 发出的查询。应用程序 140 发出的查询可被预先定义(即，被硬编码为应用程序 140 的一部分)、或者可以响应于输入(例如，用户输入)而产生。在任一种情况中，使用抽象查询接口 146 定义的逻辑字段来组成/执行查询(在这里，其被称为“抽象查询”)。具体地说，在抽象查询中使用的逻辑字段由抽象查询接口 146 的数据储存库抽象组件 148 定义。抽象查询由运行时间组件 150 执行，其中，运行时间组件 150 首先将抽象查询变换为与 DBMS 154 中包含的数据的物理表示一致的形式。

在一个实施例中，数据储存库抽象组件 148 被配置有安全信息 162。对于不是基于抽象模型(或其某些等同物)的实施例，安全信息可以驻留在别的地方。在一个实施例中，安全信息 162 包括与一个或多个字段相关的关键字(key)。下面将更详细地描述这种关键字的多个方面。

运行时间组件 150 用来进行各种分析，并且，在某些实施例中，根据所进行的分析的结果增强各种安全特征或进行其它动作。因此，运行时间组件 150 被示出为配置有实现在此描述的方法的安全算法 151(其可以是代表性的或多个算法)。通常，可以将运行时间组件 150 实现的安全特征施加到特定的用户、一组用户或全部用户。

在一个实施例中，用户通过图形用户接口(GUI)指定查询的元素。GUI 的内容由应用程序 140 产生。在特定实施例中，GUI 内容是可利用浏览器程序 122 而在客户端计算机系统 102 上呈现的超文本标记语言(HTML)内容。因此，存储器 132 包括适配成为来自客户端计算机 102 的请求服务的超文本传输协议(http)服务器进程 152(例如，网络服务器)。例如，服务器进程 152 可响应请求以访问说明性地驻留在服务器 104 上的数据库 156。到来的对来自数据库 156 的数据的客户端请求调用应用程序 140。当被处理器 130 执行时，应用程序 140 使服务器计算机 104 进行实施本发明各种方面的步骤或元素，包括访问数据库 156。在一个实施例中，应用程序 140 包括多个配置为建立随后由浏览器程序 122 呈现的 GUI 元素的小服务程序。

图 1 只是联网的客户端计算机 102 和服务器计算机 104 的一种硬件/软件配置。本发明的实施例可适用于到任何相似的硬件配置上，而不管计算机系统是复杂的多用户计算装置、单用户工作站、还是不具有其自己的非易失性存储器的网络设备。此外，应当理解，尽管参考了包括 HTML 的特定标记语言，但本发明不限于特定的语言、标准或版本。因此，本领域技术人员将认识到，本发明可适应其它标记语言以及非标记语言，并且，本发明还可适应特定标记语言未来的改变、以及目前未知的其它语言。同样，图 1 示出的 http 服务器进程 152 只是说明性的，并且已考虑到适配为支持任何已知和未知协议的其它实施例。

环境的逻辑/运行时间视图

图 2A－B 示出了本发明的组件的说明性关系视图 200。请求实体(例如，应用程序 140 之一)发出如该请求实体的各个应用程序查询规范 142 定义的查询 202。在这里，因为根据抽象(即，逻辑)字段而不是通过直接引用 DBMS 154 中的底层物理数据实体来组成所产生的查询 202，所以该查询通常被称为“抽象查询”。因此，可以定义与所使用的特定底层数据表示无关的抽象查询。在一个实施例中，应用程序查询规范 142 可既包括用于数据选择的标准(选择标

准 204)又包括基于选择标准 204 要返回的字段的明确规范(返回数据规范 206)。

在下面的表 1 中示出了对应于图 2B 示出的抽象查询 202 的说明性抽象查询。作为说明，使用 XML 来定义抽象查询 202。然而，可以有利地使用任何其它语言。

表 1 - 查询示例

```
001  <?xml version="1.0"?>
002  <!--Query string representation: (FirstName="Mary" AND LastName=
003  "McGoon") OR State="NC"-->
004  <QueryAbstraction>
005   <Selection>
006    <Condition internalID="4">
007     <Condition field ="FirstName" operator="EQ" value="Mary"
008  internalID="1"/>
009     <Condition field ="LastName" operator="EQ" value="McGoon"
010  internalID="3" relOperator="AND"></Condition>
011    </Condition>
012    <Condition field="State" operator="EQ" value="NC" internalID="2"
013  relOperator="OR"></Condition>
014  </Selection>
015  <Results>
016       <Field name="FirstName"/>
017       <Field name="LastName"/>
018       <Field name="State"/>
019   </Results>
020  </QueryAbstraction>
```

作为说明，表 1 示出的抽象查询包括含有选择标准的选择规范(第 005 - 014 行)和结果规范(第 015 - 019 行)。在一个实施例中，选择标准由(逻辑字段的)字段名、比较运算符(=、>、<等)以及(所述字段与之进行比较的)值表达式组成。在一个实施例中，结果规范是将作为查询执行结果返回的抽象字段的列表。抽象查询中的结果规范可由字段名和分类标准组成。

由应用程序查询规范 142 指定并用来组成抽象查询 202 的逻辑字段由数据储存库抽象组件 148 定义。通常，数据储存库抽象组件 148 将信息作为可在应用程序 140(其可以响应于用户输入的查询条件)发出的查询(例如，抽象查询 202)中使用的一组逻辑字段进行披露(expose)，以指定数据选择标准，并指定从查询操作返回的结果数据的形式。逻辑字段被与 DBMS 154 中使用的底层数据表示无关地定义，从而允许形成松散地结合到底层数据表示的查询。

通常，数据储存库抽象组件 148 包括统称为字段规范 208 的多个字段规范 208_1、208_2、208_3、...(作为示例，示出了 3 个字段规范)。具体地说，为可用于组成抽象查询的每个逻辑字段提供字段规范。在一个实施例中，字段规范 208 包括逻辑字段名 210_1、210_2、210_3(统称为字段名 210)和相关的访问方法 212_1、212_2、212_1(统称为访问方法 212)。

访问方法 212 将逻辑字段名关联(即，映射)到数据库(例如，数据库 156 之一)中的特定物理数据表示 214_1、214_2、..、214_N。作为说明，在图 2A 中示出了两种数据表示，即 XML 数据表示 214_1 和关系数据表示 214_2。然而，物理数据表示 214_N 表明考虑了任何其它已知或未知的数据表示。

在一个实施例中，单个数据储存库抽象组件 148 包含用于两种或更多物理数据表示 214 的字段规范(和相关的访问方法)。在替换实施例中，为每个单独的物理数据表示 214 提供不同的单个数据储存库抽象组件 148。在另一实施例中，提供多个数据储存库抽象组件 148，其中，每个数据储存库抽象组件 148 披露同一底层物理数据(其可包括一个或多个物理数据表示 214)的不同部分。以这一方式，多个用户可同时使用单个应用程序 140，来访问同一底层数据，其中，由各个数据储存库抽象组件 148 确定向应用程序披露的底层数据的特定部分。

根据要支持的不同类型逻辑字段的数目，构思任何数目的访问方法。在一个实施例中，提供了针对简单字段、过滤字段和组合字段的访问方法。字段规范 208_1、208_2 和 208_3 分别例示了简单字段访问方法 212_1、212_2 和 212_3。简单字段被直接映射到底层物理数据表示中的特定实体(例如，映射到给定数据库表和列的字段)。作为说明，图 2B 示出的简单字段访问方法 212_1 将逻辑字段名 210_1(“FirstName(名)”)映射到名称为 “contact(联系)” 的表中名称为 “f_name” 的列。过滤字段(图 2 中未示出示例)标识相关的物理实体，并提供用来定义该物理数据表示内的特定条目子集的规则。过滤字段的示例是纽约

邮政编码字段，该字段映射到邮政编码的物理表示，并将数据限制为仅仅为纽约州定义的那些邮政编码。组合访问方法(在图 2 中未示出示例)使用作为访问方法定义的一部分提供的表达式来从一个或多个物理字段计算逻辑字段。以这一方式，可以计算出底层数据表示中不存在的信息。一个示例是通过将销售价格字段乘以营业税率而组成的营业税字段。

已经考虑到底层数据的任何给定数据类型(例如，日期、十进制数等)的格式可能变化。因此，在一个实施例中，字段规范 208 包括反映底层数据格式的类型属性。然而，在另一实施例中，字段规范 208 的数据格式与相关底层物理数据不同，在此情况下，访问方法负责以请求实体采取的适当格式返回数据。这样，该访问方法必须知道采取了哪种数据格式(即，根据逻辑字段)、以及底层物理数据的实际格式。然后，该访问方法可以将底层物理数据转换为逻辑字段的格式。

作为示例，图 2 示出的数据储存库抽象组件 148 的字段规范 208 代表被映射到关系数据表示 214_2 中表示的数据的逻辑字段。然而，数据储存库抽象组件 148 的其它实例将逻辑字段映射到其它物理数据表示，例如 XML。

在一个实施例中，一个或多个字段规范 208 配置有在上面参考图 1 简要描述的安全信息 162。在所示出的实施例中，仅字段定义 208_3 具有相关联的安全信息。因此，应当理解，不是所有字段定义都必定需要包括安全信息。在本示例中，安全信息是具有值“关键字”的类型属性 220。应当理解，关键字值不需要在数据储存库抽象 148 中指定，而是可以是例如配置文件中的值。在操作中，为具有关键字并且用户已经将其包括在至少一个查询中的每个字段保存会话特定列表 153(在图 1 中示出了多个会话特定列表 153)。具体地说，列表 153(例如，散列表)包含已经从用于特定会话的相关字段返回的所有值。因此，通常，对于返回先前未返回过的结果(即，不重叠的查询结果)的每个查询，用于给定用户的尺寸列表增大。在一个实施例中，该列表可以是持久的，而在另一实施例中，当用户退出时或在用户一段时间不活动之后删除该列表。然后，如将在下面更详细地描述的那样，可进行查询结果分析。在某些情况下，根据安全动作定义 213 来进行动作。下面描述说明性动作。

表 II 示出对应于图 2B 所示的数据储存库抽象组件 148 的说明性数据储存库抽象组件。作为说明，使用 XML 来定义数据储存库抽象 148。然而，可以有利地使用任何其它语言。

表 II - 数据储存库抽象示例

```
<?xml version="1.0"?>
<DataRepository>
 <Category name="Demographic">
  <Field queryable="Yes" name="FirstName" displayable="Yes">
      <AccessMethod>
        <Simple columnName="f_name" tableName="contact"></Simple>
      </AccessMethod>
      <Type baseType="char"></Type>
  </Field>
  <Field queryable="Yes" name="LastName" displayable="Yes">
      <AccessMethod>
        <Simple columnName="l_name" tableName="contact"></Simple>
      </AccessMethod>
      <Type baseType="char"></Type>
  </Field>
  <Field queryable="Yes" name="Clinic Number" displayable="Yes">
      <AccessMethod>
        <Simple columnName="CN" tableName="contact"></Simple>
      </AccessMethod>
      <Type baseType="char" key="true"></Type>
      <Security>
        <SecurityRule>
          <User>All</User>
          <Action> RunAndLog</Action>
        </SecurityRule>
        <SecurityRule>
          <User> securityOfficers </User>
          <Action> RunAndLog </Action>
        </SecurityRule>
        <SecurityRule>
```

```
                <User> cujo </User>
                <Action> NoAction </Action>
            </SecurityRule>
        </Security>
    </Field>
  <Category>
</DataRepository>
```

图 3 示出了例示运行时间组件 150 的操作的一个实施例的说明性运行时间方法 300。当运行时间组件 150 接收到抽象查询(例如图 2 示出的抽象查询 202)的实例作为输入时，在步骤 302 进入方法 300。在步骤 304，运行时间组件 150 读取并分析该抽象查询实例，并定位单独的选择标准和所希望的结果字段。如下所述，在步骤 309，进行将有利地与执行后结果分析一起使用的一些的初步语句结构分析。具体地说，在步骤 309，计算查询共同性(commonality)值。该查询共同性值通过确定当前查询和所有先前查询之间的相对共同性来计算。例如，如果一个查询具有两个条件，即门诊号码>x 且邮政编码 = y，而同一用户的另一查询具有两个条件，即门诊号码<1000 且诊断结果 = z，则这两个查询具有 50%的共同性。

在步骤 306，运行时间组件 150 进入用于处理在抽象查询中出现的每个查询选择标准语句的循环，从而构建具体查询的数据选择部分。在一个实施例中，选择标准(在这里也称为条件)由(逻辑字段的)字段名、比较运算符(=、>、<等)和将字段与之相比较的值表达式组成。在步骤 308，运行时间组件 150 使用来自抽象查询的选择标准的字段名来在数据储存库抽象 148 中查找该字段的定义。如上所述，该字段定义包括对用来访问与该字段相关联的物理数据的访问方法的定义。

在步骤 310 处开始之后，进行更多步骤来进行语句结构分析。具体地说，在步骤 310，进入用于每个先前查询的循环。也就是说，访问并遍历查询历史表 157(图 1)。通常，查询历史表 157 是已经执行过的查询的列表。每当执行新查询时，在查询历史表 157 中填充新条目。在一个实施例中，此数据结构包含处于其抽象形式的 SQL 查询。可以对该数据结构配置释放历史的时间。用于释放历史的一个选择是在结束会话的时候。另一个是在经过一定的时间段之后。在步骤 312，运行时间组件 150 确定在步骤 310 从历史查询表 157

中检索到的先前的查询中是否使用了正被处理(步骤 306)的查询选择的字段。如果未使用，则方法 300 返回步骤 310，并且运行时间组件 150 从历史查询表 157 中检索另一先前查询。当识别出具有正被处理(步骤 306)的查询选择字段的先前查询时，对该查询选择和所识别的先前查询进行分析(步骤 314)。在步骤 316，运行时间组件 150 确定分析结果(在步骤 314)是否需要进行某个动作。在一个实施例中，在数据储存库抽象组件 148 中指定所述动作(参见表 II)。安全动作包括记录用户的查询(或其它相关信息)、防止该查询被执行、和/或终止用户的会话。更一般地，本领域技术人员将认识到，当调用安全规则时，可以进行任何种类的响应。例如，可以向系统管理员发出通知(例如，通过电子邮件)。注意，在表 II 示出的示例中，为单个用户(例如，Cujo)、多组用户(例如，安全官员)和所有用户定义了安全动作。在对于特定字段存在多个动作的一个实施例中，适用为用户最紧密地设计的动作。这样，特定于单个用户的动作优先于所有其它动作，并且特定于一组的动作优先于为所有用户指定的动作。为所有用户指定的动作仅在不存在其它更紧密地为用户设计的动作时才适用。如果步骤 314 的回答是否定的(即，不需要动作)，则处理返回步骤 310，其中从历史查询表 157 中检索另一先前查询以供检查。如果在步骤 316 需要动作，则在步骤 318 进行该动作。如果该动作是致命的(fatal)(步骤 320)，则不执行用户的查询(步骤 322)。否则，处理返回步骤 310。一旦对历史查询表 157 中的每个先前查询都已经检查了正被处理的当前查询选择的字段的存在性，方法 300 就前进到步骤 324。

然后，运行时间组件 150 为正被处理的逻辑字段建立(步骤 324)具体查询组分(contribution)。如同在这里定义的那样，具体查询组分是用来基于当前逻辑字段进行数据选择的具体查询的一部分。具体查询是采用例如 SQL 和 XML 查询的语言表示的查询，并与给定物理数据储存库(例如，关系数据库或 XML 储存库)的数据一致。因此，使用具体查询来从由图 1 示出的 DBMS 154 表示的物理数据储存库中定位和检索数据。然后，将为当前字段产生的具体查询组分添加到具体查询语句中。然后，方法 300 返回步骤 306，以便开始对抽象查询的下一字段的处理。因此，对抽象查询中的每个数据选择字段重复在步骤 306 进入的过程，从而贡献出(contribute)要进行的最终查询的附加内容。

在构建具体查询的数据选择部分之后，运行时间组件 150 识别将作为查询执行的结果返回的信息。如上所述，在一个实施例中，抽象查询定义在这

里称为结果规范的、将作为查询执行结果而返回的抽象字段的列表。抽象查询中的结果列表可包括字段名和分类标准。因此，方法 300 在步骤 328 处进入循环(由步骤 328、330、332 和 334 定义)，以便将结果字段定义添加到所产生的具体查询中。在步骤 330，运行时间组件 150 在数据储存库抽象 148 中查找(来自抽象查询的结果规范的)结果字段名，然后从数据储存库抽象 148 中检索结果字段定义，以便识别将为当前逻辑结果字段返回的数据的物理位置。然后，运行时间组件 150 为该逻辑结果字段构建(如步骤 332)(识别要返回的数据的物理位置的具体查询的)具体查询组分。然后，在步骤 334，将具体查询组分添加到具体查询语句中。一旦处理了抽象查询中的每个结果规范，就在步骤 336 执行该查询。

参考图 4 来描述用于根据步骤 310 和 318 为逻辑字段构建具体查询组分的方法 400 的一个实施例。在步骤 402，方法 400 查询与当前逻辑字段相关的访问方法是不是简单访问方法。如果是，则基于物理数据位置信息构建(步骤 404)具体查询组分，随后，处理根据上述方法 300 而继续进行。否则，处理继续进行到步骤 406，以查询与当前逻辑字段相关的访问方法是否是过滤访问方法。如果是，则基于某个物理数据实体的物理数据位置信息来构建(步骤 408)具体查询组分。在步骤 410，利用用于对与该物理数据实体相关的数据进行子集构造(subset)的附加逻辑(过滤器选择)来扩展具体查询组分。然后，处理根据上述方法 300 而继续进行。

如果该访问方法不是过滤访问方法，则处理从步骤 406 前进到步骤 412，在步骤 412，方法 400 查询访问方法是不是组合访问方法。如果该访问方法是组合访问方法，则在步骤 414，定位并检索组合字段表达式中每个子字段引用(reference)的物理数据位置。在步骤 416，用组合字段表达式的物理字段位置信息代替该组合字段表达式的逻辑字段引用，由此产生具体查询组分。然后，处理根据上述方法 300 而继续进行。

如果该访问方法不是组合访问方法，则处理从步骤 412 前进到步骤 418。步骤 418 代表作为本发明的实施例而构思的任何其它访问方法类型。然而，应当理解，构思了并不实现所有可用的访问方法的实施例。例如，在特定实施例中，仅使用简单访问方法。在另一实施例中，仅使用简单访问方法和过滤访问方法。

如上所述，如果逻辑字段指定了与底层物理数据不同的数据格式，则可

能有必要进行数据转换。在一个实施例中，当根据方法 400 建立逻辑字段的具体查询组分时，对于每一个访问方法，分别进行初始转换。例如，可以作为步骤 404、408 和 416 的一部分或紧跟其后进行转换。在步骤 322 执行了查询之后，进行随后从物理数据的格式到逻辑字段的格式的转换。当然，如果逻辑字段定义的格式与底层物理数据相同，则没有必要进行转换。

参考图 5，示出了方法 500，其说明在步骤 314 进行的分析的一个实施例。回想(recall)对具有通用格式<字段><运算符><值>的选择/条件进行分析。在步骤 502，使用运算符和值来确定查询选择覆盖的范围。在步骤 504，运行时间组件 150 检查相对于在步骤 310 中从历史查询表 157 检索的先前查询的条件的不重叠条件。在一个实施例中，定义不重叠条件为具有早先查询的共同字段但不返回早先查询所返回的任何结果(行)的条件。例如，考虑具有范围条件“年龄≥0 且年龄<5”的先前查询(其条件存储在历史查询表 157 中)。现在假设正被分析的查询包含范围条件“年龄≥5 且年龄<10”。这些查询条件显示(evidence)了这样的模式，该模式暗示：用户正通过有意识地构造设计用来避免返回任何相同行的查询而扫描数据库的大部分。在另一实施例中，将不重叠条件定义为具有早先查询的共同字段并返回一些新结果(即，先前查询未返回过的结果)和一些旧结果(即，先前查询已返回过的结果)的条件。这种不重叠条件的重复模式也可被识别为访问/积累数据库的一部分的未经授权的企图。

如果识别出不重叠条件，则在步骤 316/318 处理该条件。在一个实施例中，根据管理员设置来处理不重叠条件。具体地说，可以通过管理员设置来指定在进行某个动作之前必须识别的不相关查询的数目。此外，一个实施例可允许一定程度的条件重叠或分离。因而，仍然可以将具有一些很小数目的共同结果的两个查询之间的条件视为不重叠。在这种情况下，基于不同查询的条件所覆盖的范围来确定不重叠可令人满意。例如，在具有相关字段的某组查询的结果的总范围是 4000，而将由条件返回的重叠结果的实际数目是 4 的情况下，查询/条件基本上不重叠。另一方面，在具有相关字段的某组查询的结果的总范围是 40，而将由查询返回的重叠结果的数目是 30 的情况下，查询/条件可被认为是基本上重叠的。出于权利要求解释(construction)的目的，术语“不重叠”查询/条件应被解释为包括基本上不重叠的查询/条件。此外或者可替换地，可以通过管理员设置来定义可为之返回结果的不同病人的数目。

在一个实施例中，可以使这种管理员设置特定于特定用户。因而，可以给第一用户更多的数据访问权，而可以相对多地限制第二用户的访问权。

上文例示了执行前分析。其它或替换方面包括图 3 的步骤 336 处的查询执行之后的执行后分析。说明性的执行后分析由块 338、340 和 342 代表。通常，执行后分析包括在执行查询之后以及在将所执行的查询的结果返回给用户之前或/和之后进行的处理。例如，块 338 代表在将结果提供给用户之前进行的不重叠查询分析。图 6 示出了用于进行块 338 的不重叠查询分析的方法 600 的一个实施例。首先，在步骤 602，运行时间组件 150 进入为结果的每一列进行的循环。在步骤 604，运行时间组件 150 确定该列是不是关键字列(key column)(即，为其定义了关键字的列)。如果不是，则类似地处理结果的下一列。如果结果不包括关键字列，则检索对应于该关键字列的列表 153 的当前尺寸(步骤 606)。将尚未包含在列表 153 中的结果中的每个值添加到列表 153 中(步骤 608)。在步骤 610，运行时间组件 150 确定是否识别出不重叠查询。在说明性实施例中，步骤 610 包括确定添加每个新值(步骤 608)之后的关键字列表(key list)的尺寸是否等于新结果/值的数目与(在步骤 606 检索到的)该列表的原始尺寸之和。在这方面，肯定性的确定表明查询没有返回新值，并且没有将新值添加到列表 153 中(在此情况中，在步骤 336 执行的查询不与先前查询重叠)。

如先前对执行前分析所描述的那样，在某些情况中，仍然可以将一定程度的重叠视为基本上不重叠。这一原理可以适用于执行后分析。因而，仍然可以将具有一些很小数目的共同结果的两个查询之间的结果视为不重叠。在这种情况中，基于返回结果的总数目来确定不重叠可令人满意。例如，在具有相关字段的某组查询的全部结果是 4000 个、而重叠结果的数目是 4 的情况中，查询基本上不重叠。另一方面，在具有相关字段的某组查询的全部结果是 40 个，而重叠结果的数目是 30 的情况中，可以认为该查询基本上是重叠的。出于权利要求解释的目的，术语“不重叠”查询/结果应当被解释成包括“基本上”不重叠的查询/结果。如果在步骤 336 执行的查询被确定为重叠或基本上重叠，则对结果进行标记(步骤 611)，以便返回给用户，并且，对下一列继续进行处理。否则，运行时间组件 150 确定(步骤 614)是否需要某个预先定义的动作(已经在上面描述了它的示例)。如果需要，则在步骤 616 进行该动作。如果该动作是致命的(在步骤 618 确定)，则终止请求，并且不把结果返回

给用户(步骤 620)。然后，方法 600 退出。如果该动作不是致命的，则处理返回步骤 602，在步骤 602，开始对下一列的处理。如果在没有调用致命动作的情况下成功地处理了所有列，则在步骤 612 将所有结果返回给用户。

作为用来识别不重叠查询的执行后查询分析的示例，考虑执行返回 1000 个不同的门诊号码的第一查询的用户。在适当的门诊号码的关键字列表 153 中，跟踪这 1000 个不同的门诊号码。然后，用户执行返回 1500 个不同的门诊号码的第二查询。假设第一查询和第二查询返回完全唯一的结果，则门诊号码的关键字列表 153 将包含 2500 个不同的门诊号码，并且，查询被确定为不重叠。如果查询返回的结果共享至少一个共同的值，则可以进行步骤以确定查询是否仍然基本上不重叠(如上所述)。更一般地，可采用任何种类的可配置设置来确定不重叠查询的模式，并避免过早的致命动作(即，防止将结果返回给用户)。例如，可以预先定义可在进行动作之前返回的不重叠关键字值的数目。可替换地或者此外，可以预先定义可在进行动作之前执行的不重叠或基本不重叠的查询的数目。本领域技术人员将认识到可有利地使用的其它规则。

应当注意，使用预先定义的关键字仅仅是用于进行各种类型的查询分析的一个实施例。更一般地，考虑了允许跟踪查询之间的共同性的任何途径。例如，对预先定义的关键字的替换是由同一用户检查一系列查询以确定共同字段的存在性。然后，可以指定该共同字段，并将其用作可以通过其进行趋势分析(例如，确定不重叠查询)的关键字。

另一种执行后查询分析由图 3 中的方框 340 代表，并且在这里被称为查询联合分析检测。已经在上面提供了查询联合分析的示例。通常，查询联合分析检测检查查询序列，并确定明显不关联(即，由不同的条件组成)的查询的模式，其中，所述明显不关联的查询在递减的结果集合中仍然包含一个或多个共同结果值。用于检测和处理查询联合分析的一个实施例是图 7 示出的方法 700，其中，在执行查询之后进入该方法 700。在步骤 702，安全算法 151 确定是否存在用于跟踪查询结果的结果列表。如果不存在，则创建结果列表 161，并将结果存储在其中(步骤 704)。然后，方法 700 退出。然而，如果结果列表已经存在，则算法 151 进行操作，以便从结果列表 161 中丢弃所有非共同的值。也就是说，从结果列表 161 中移除包含在结果列表 161 中的不是从执行所述查询返回的结果的一部分的所有值。在步骤 708，算法 151 确定

结果列表的尺寸是否已经降到尺寸阈值之下(其中，在一个实施例中，尺寸阈值是可定制的)。如果不是，则将结果返回给用户(步骤 710)，并且方法 700 退出。否则，算法 151 确定(在图 3 的步骤 305 确定的)共同性值是否小于共同性值阈值(步骤 712)。如果不小于，则将结果返回给用户(步骤 710)，并且方法 700 退出。否则，在步骤 714 进行预先定义的安全动作。如果该安全动作是致命的(在步骤 716 确定)，则停止用户的请求，并且方法 700 退出。如果该安全动作不是致命的，则将结果返回给用户(步骤 710)，并且方法 700 退出。

另一类型的执行后查询分析由图 3 中的块 342 代表，并且在这里被称为削减分析检测。削减分析是指这样的过程：执行返回相对大数目的行的宽泛查询，随后持续且系统地利用后续查询对初始结果进行子集构造。在一个方面，削减分析是联合查询分析的变体；两种途径都有利地使用用户已知的信息来限制返回结果的尺寸。考虑发出对患有 Alzheimer 氏病的人的第一查询的用户。在查看通过执行第一查询返回的结果之后，用户确定可以通过将查询限制为居住在 California 的那些人来获得更大程度的确切性。因此，用户发出对患有 Alzheimer 氏病并且居住在 California 的人的第二查询。随后，用户将查询进一步限制为特定年龄的人。用户可以通过任何数目的查询来继续此子集构造的模式，以便减小返回结果的数目。

图 8 示出了执行后削减检测方法 800 的一个实施例，其中，在执行查询并接收到结果之后进入该方法 800。在步骤 804，运行时间组件 150 确定结果计数是否小于跟踪阈值。作为说明，跟踪阈值是根据应当何时进行削减检测而选择的预先定义的值。也就是说，如果结果计数大于跟踪阈值，则不进行削减检测，以便赋予用户一定程度的搜索能力。因此，如果步骤 804 的回答是否定的，则将查询执行的结果返回给用户(步骤 806)。然而，如果结果计数小于跟踪阈值，则运行时间组件 150 根据削减检测方法的前一次调用确定是否已经存在一个或多个结果列表 161(图 1)。通常，结果列表 161 包含出于进行削减检测的目的而执行的查询的结果。如果(在步骤 808)结果列表尚不存在，则将当前结果存储在结果列表 161 中(步骤 810)，并随后将其返回给用户(步骤 806)。如果存在至少一个结果列表 161，则运行时间组件 150 确定当前结果是不是任何一个现有结果列表的子集(步骤 812)。如果不是，则将当前结果存储在单独的结果列表中(步骤 814)。因此，可存在多个结果列表，每个结果列表包含为不同查询返回的多组不相关结果。然而，如果当前结果是现有

结果列表之一的子集，则已经检测到削减模式，并调用安全动作(步骤 816)。已经在上面描述了说明性的安全动作。如果该安全动作是致命的(在步骤 818 确定)，则不把当前结果返回给用户，从而可以防止用户执行任何进一步的查询(步骤 820)。如果该安全动作不是致命的，则可以将结果返回给用户(步骤 806)。

在上述削减方法 800 中，假设两个查询的结果计数小于跟踪阈值(在步骤 804 确定)，则可以在仅仅两个查询之后检测到削减模式。然而，应当理解，用于检测削减模式的特定标准是可配置的。例如，削减算法(除了小于跟踪阈值的结果计数以外)可能需要削减模式跨越一定数目 N 个查询，其中 N>2。此外，削减算法可能需要削减模式通过顺序/连续查询而发生。本领域技术人员将认识到可有利地使用的其它标准。

在一个实施例中，有利地使用“热门列表”。热门列表包含所选择的应得到更高安全等级的个人。在一个实施例中，对所有查询使用单个热门列表，而不管用户是谁。在热门列表中列出的个人是名人的情况下，这种途径可能是有用的。在另一实施例中，将热门列表对每个用户进行个人化，使得该列表包含各个用户认识的个人。以这一方式，可以检测并处理由特定用户进行的指向该用户的热门列表上的一个或多个个人的搜索，以保护匿名性和机密性。

如上所述，数据储存库抽象组件 148 仅仅说明提供各种优点的一个实施例。在一个方面，通过定义应用程序查询规范和底层数据表示之间的松散耦合来获得多个优点。如同使用 SQL 的情况那样，不是利用特定的表、列和关系信息来对应用程序编码，而是该应用程序以更加抽象的方式定义数据查询需求，该数据查询需求随后在运行时被绑定到特定的物理数据表示。本发明松散的查询－数据耦合使得请求实体(例如应用程序)，即使在底层数据表示被修改、或者将与全新的而不是在开发该请求实体时使用的物理数据表示一起使用该请求实体时，也能起作用。在修改或重新构造给定物理数据表示的情况下，更新对应的数据储存库抽象，以反映对底层物理数据模型进行的改变。同一组逻辑字段可通过查询获得以供使用，并且其仅仅被绑定到物理数据模型中的不同实体或位置。因此，即使对应的物理数据模型已经经历了重大改变，写入到抽象查询接口的请求实体也继续不被改变地运行。如果将与全新的而不是在开发请求实体时使用的物理数据表示一起使用该请求实体，则可

以使用相同的技术(例如，关系数据库)但遵循不同的命名和组织信息策略(例如，不同大纲)来实现该新物理数据模型。新大纲将包含可使用简单、过滤和组合字段访问方法技术映射到应用程序所需的逻辑字段组的信息。可替换地，新物理表示可使用用于表示类似信息的替换技术(例如，相对于关系数据库系统而使用基于 XML 的数据储存库)。在任一种情况下，当为其提供替换数据储存库抽象时，被写入以使用抽象查询接口的现有请求实体可容易地移植，以使用新物理数据表示，该替换数据储存库抽象将查询中引用的字段与新物理数据模型中的位置和物理表示相映射。

对于最终用户，数据储存库抽象提供了披露相关数据并防止访问选定内容的数据过滤机制。然而，应当理解，数据储存库抽象仅仅是本发明的一个实施例。更一般地，根据用户数据的相关性，按照提供对查询的执行(或不执行)的方式来实施本发明。也就是说，使查询执行依赖于最终用户和在执行时将由该查询访问/返回的特定数据。

然而，应当强调的是，本领域技术人员将容易地认识到，可以与数据储存库抽象组件相分离地实现本发明的安全特征和机制。例如，在传统关系数据库的环境中，一个实施例使用来自查询分析器的结构，该结构将驻留在数据库引擎中，以便执行在此描述的分析。

尽管前述内容针对的是本发明的实施例，但是在不脱离本发明的基本范围的情况下，可以设计本发明的其它和另外的实施例，而本发明的范围由所附权利要求确定。

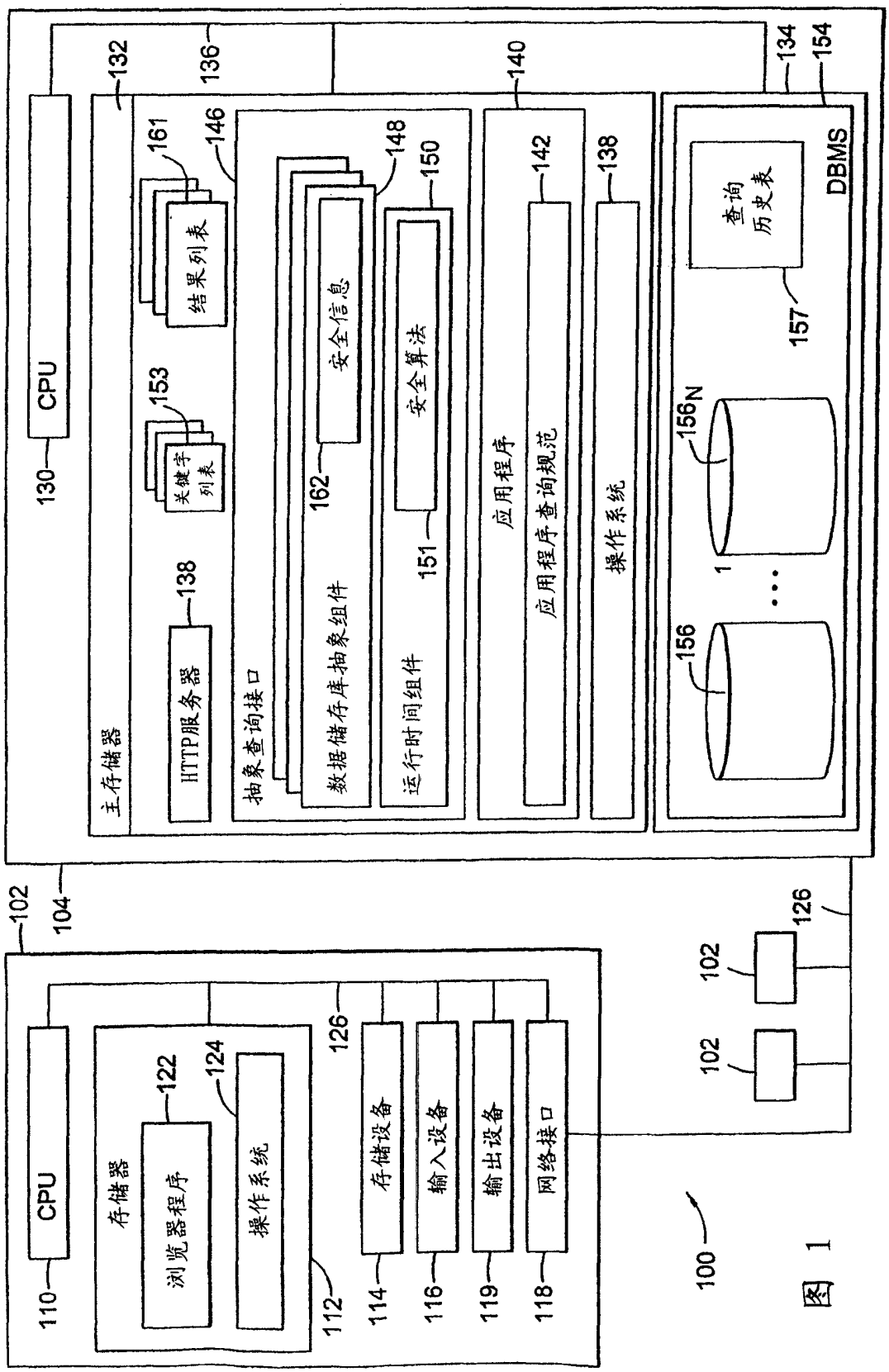

CPU
130
主存储器
132
HTTP服务器
138
抽象查询接口
136
结果列表
161
146
关键字列表
153
数据储存库抽象组件
162
安全信息
148
运行时间组件
151
安全算法
150
应用程序
140
应用程序查询规范
142
操作系统
138
DBMS
154
查询历史表
157
156N
156
1
...
102
104
126
CPU
110
存储器
浏览器程序
122
操作系统
124
112
存储设备
114
输入设备
116
输出设备
119
网络接口
118
100

图 1

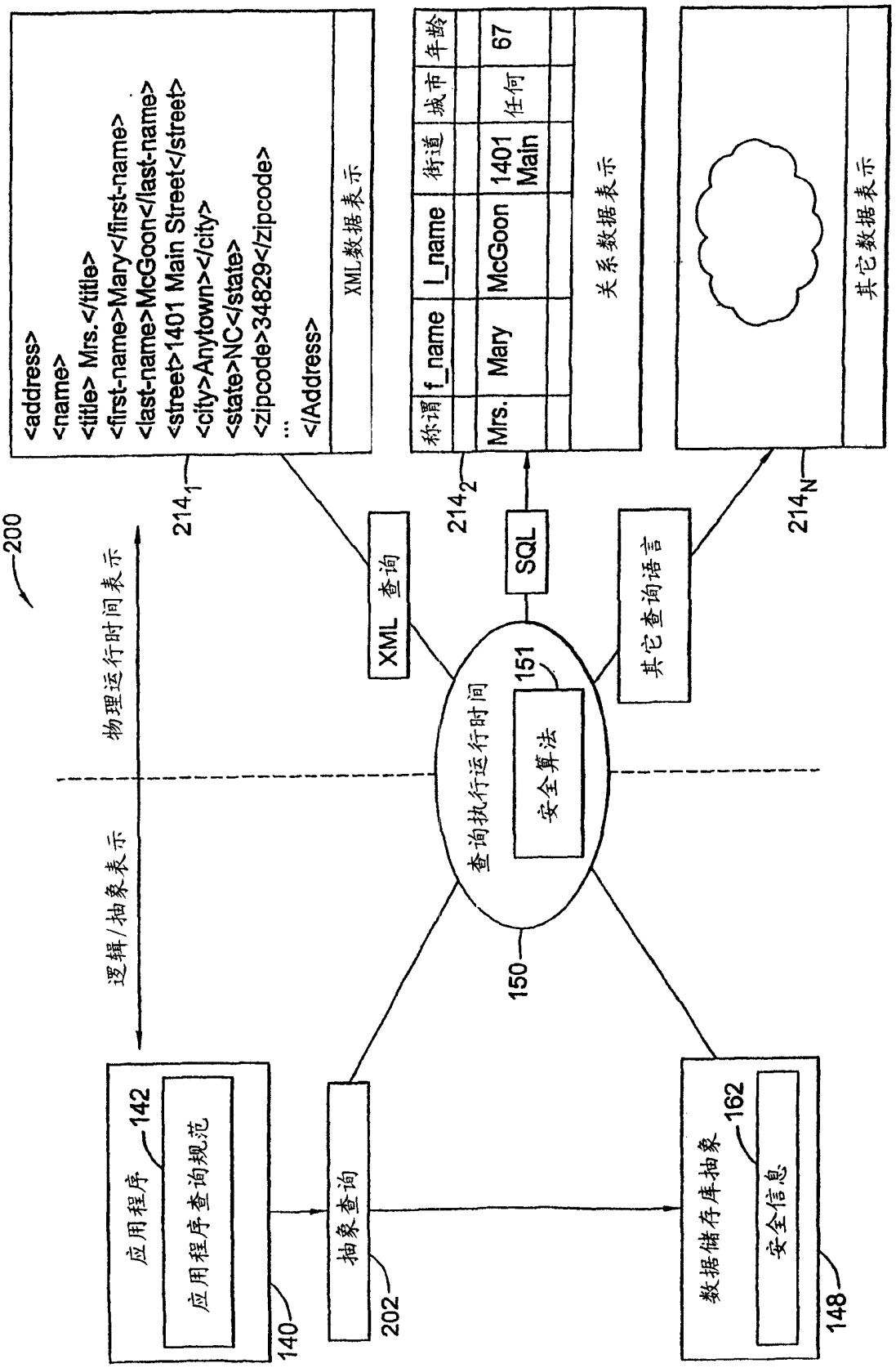

200
物理运行时间表示
逻辑/抽象表示
应用程序
142
应用程序查询规范
140
抽象查询
202
查询执行运行时间
151
安全算法
150
XML 查询
SQL
其它查询语言
数据储存库抽象
162
安全信息
148
<address>
<name>
<title> Mrs.</title>
<first-name>Mary</first-name>
<last-name>McGoon</last-name>
<street>1401 Main Street</street>
<city>Anytown></city>
<state>NC</state>
<zipcode>34829</zipcode>
...
</Address>
XML数据表示
2141
称谓 f_name l_name 街道 城市 年龄
Mrs. Mary McGoon 1401 Main 任何 67
关系数据表示
2142
其它数据表示
214N

图 2A

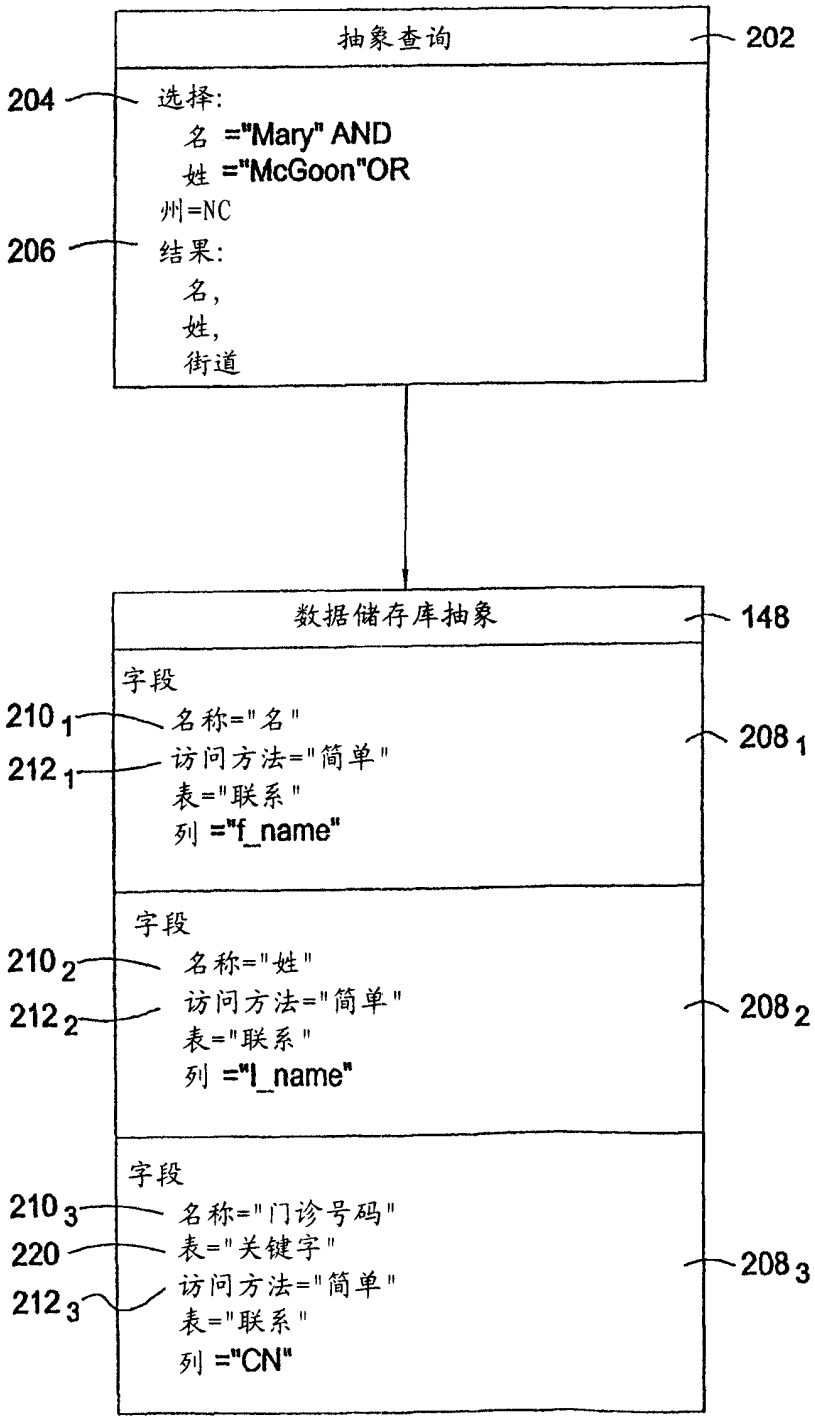

抽象查询
202
204
选择:
名 ="Mary" AND
姓 ="McGoon"OR
州=NC
206
结果:
名,
姓,
街道
数据储存库抽象
148
字段
210 1
名称="名"
212 1
访问方法="简单"
表="联系"
列 ="f_name"
208 1
字段
210 2
名称="姓"
212 2
访问方法="简单"
表="联系"
列 ="l_name"
208 2
字段
210 3
名称="门诊号码"
220
表="关键字"
212 3
访问方法="简单"
表="联系"
列 ="CN"
208 3

图 2B

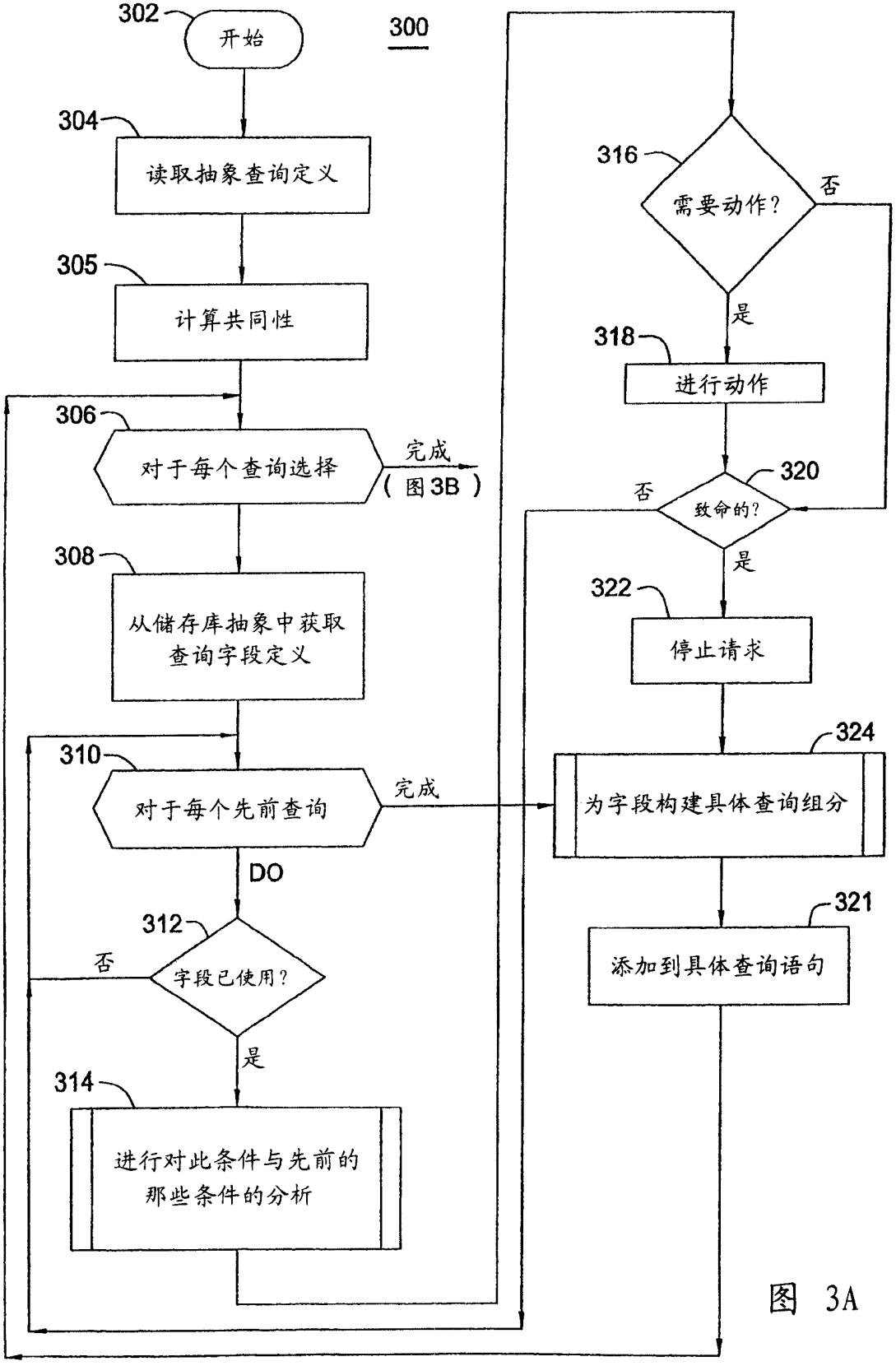

302
开始
300
304
读取抽象查询定义
305
计算共同性
306
对于每个查询选择
完成
(图3B)
308
从储存库抽象中获取
查询字段定义
310
对于每个先前查询
完成
DO
312
字段已使用?
否
是
314
进行对此条件与先前的
那些条件的分析
316
需要动作?
否
是
318
进行动作
320
致命的?
否
是
322
停止请求
324
为字段构建具体查询组分
321
添加到具体查询语句

图 3A

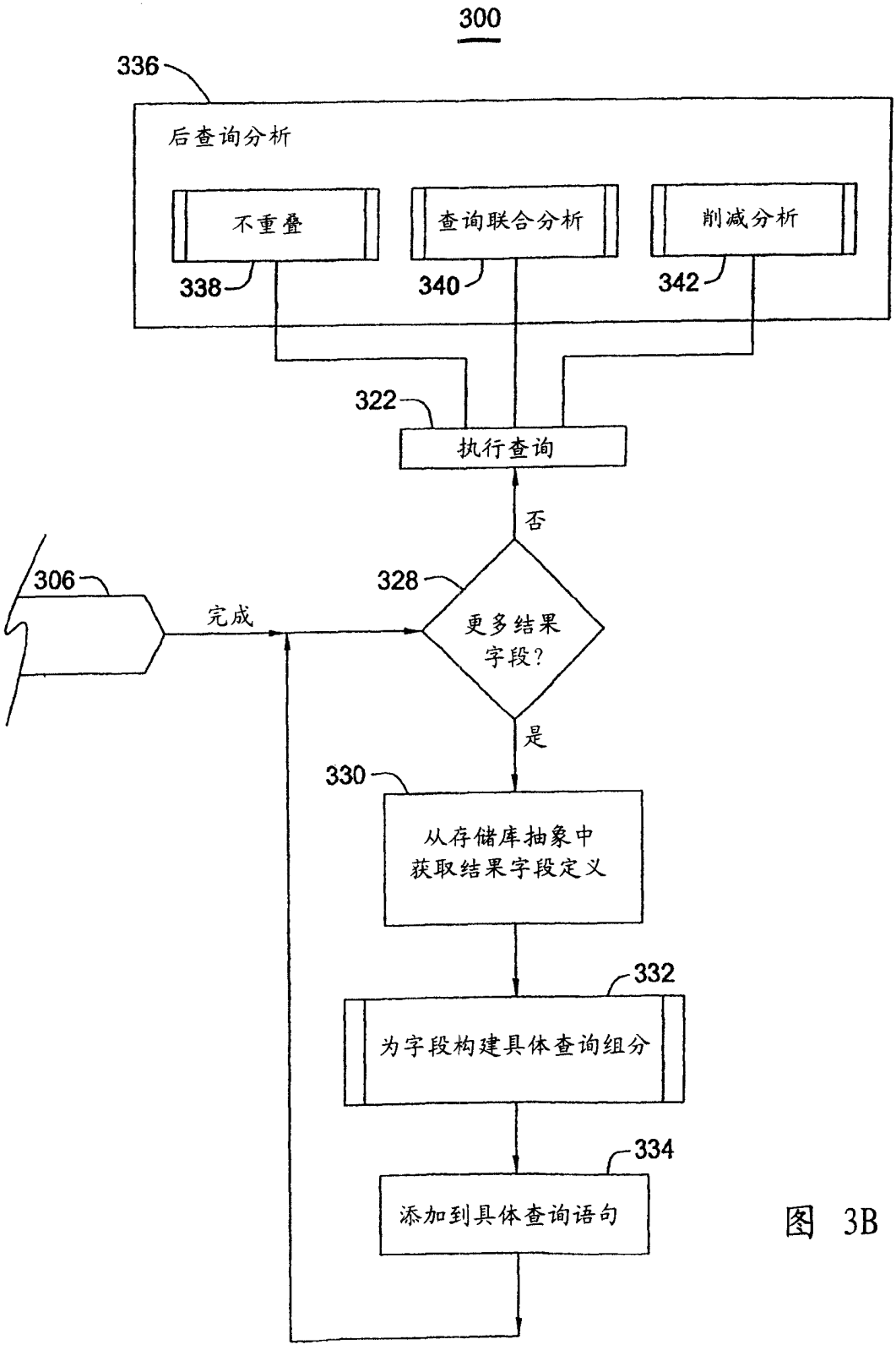

300
336
后查询分析
不重叠
查询联合分析
削减分析
338
340
342
322
执行查询
否
306
328
完成
更多结果字段?
是
330
从存储库抽象中获取结果字段定义
332
为字段构建具体查询组分
334
添加到具体查询语句

图 3B

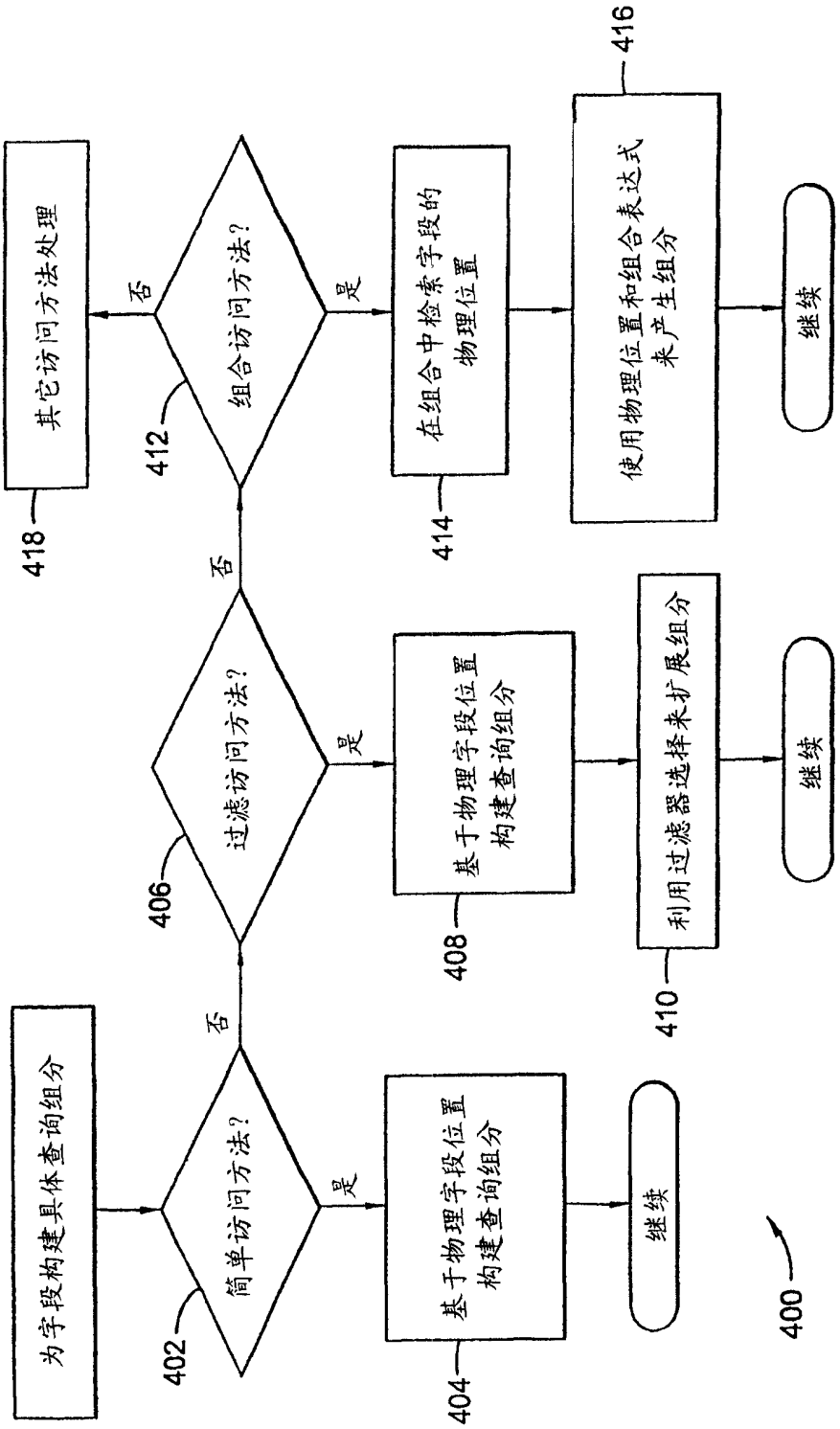

为字段构建具体查询组分
简单访问方法?
是
否
基于物理字段位置构建查询组分
继续
过滤访问方法?
是
否
基于物理字段位置构建查询组分
利用过滤器选择来扩展组分
继续
组合访问方法?
是
否
在组合中检索字段的物理位置
使用物理位置和组合表达式来产生组分
继续
其它访问方法处理
402
404
406
408
410
412
414
416
418
400

图 4

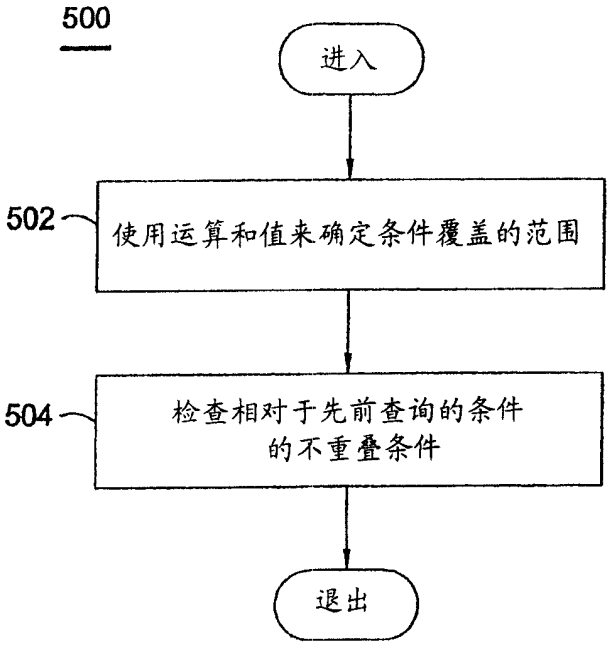


图 5

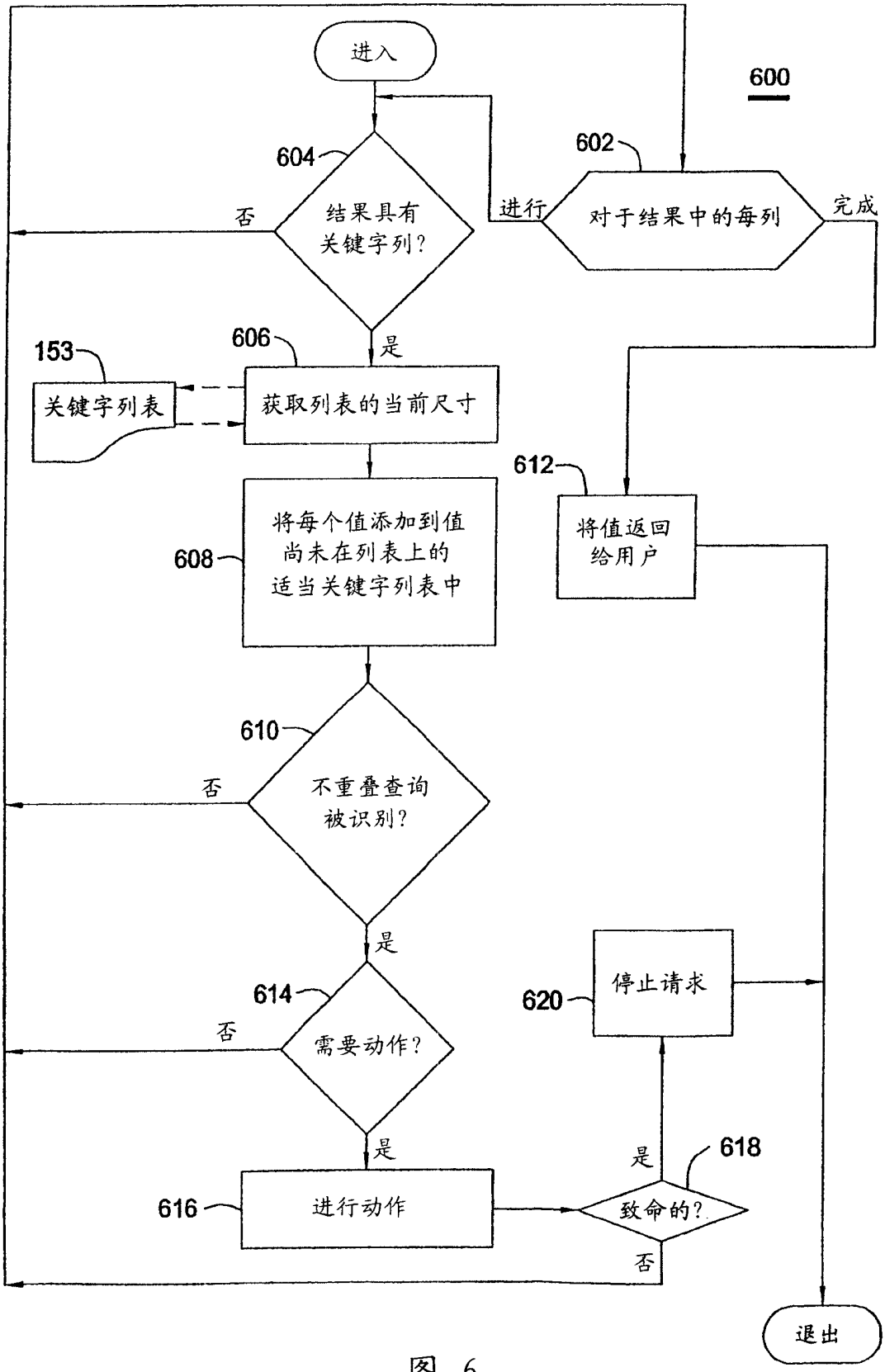

进入
600
602
604
结果具有关键字列？
否
进行
对于结果中的每列
完成
是
153
606
关键字列表
获取列表的当前尺寸
612
将每个值添加到值尚未在列表上的适当关键字列表中
608
将值返回给用户
610
不重叠查询被识别？
否
是
614
620
停止请求
需要动作？
否
是
是
618
616
进行动作
致命的？
否
退出

图 6

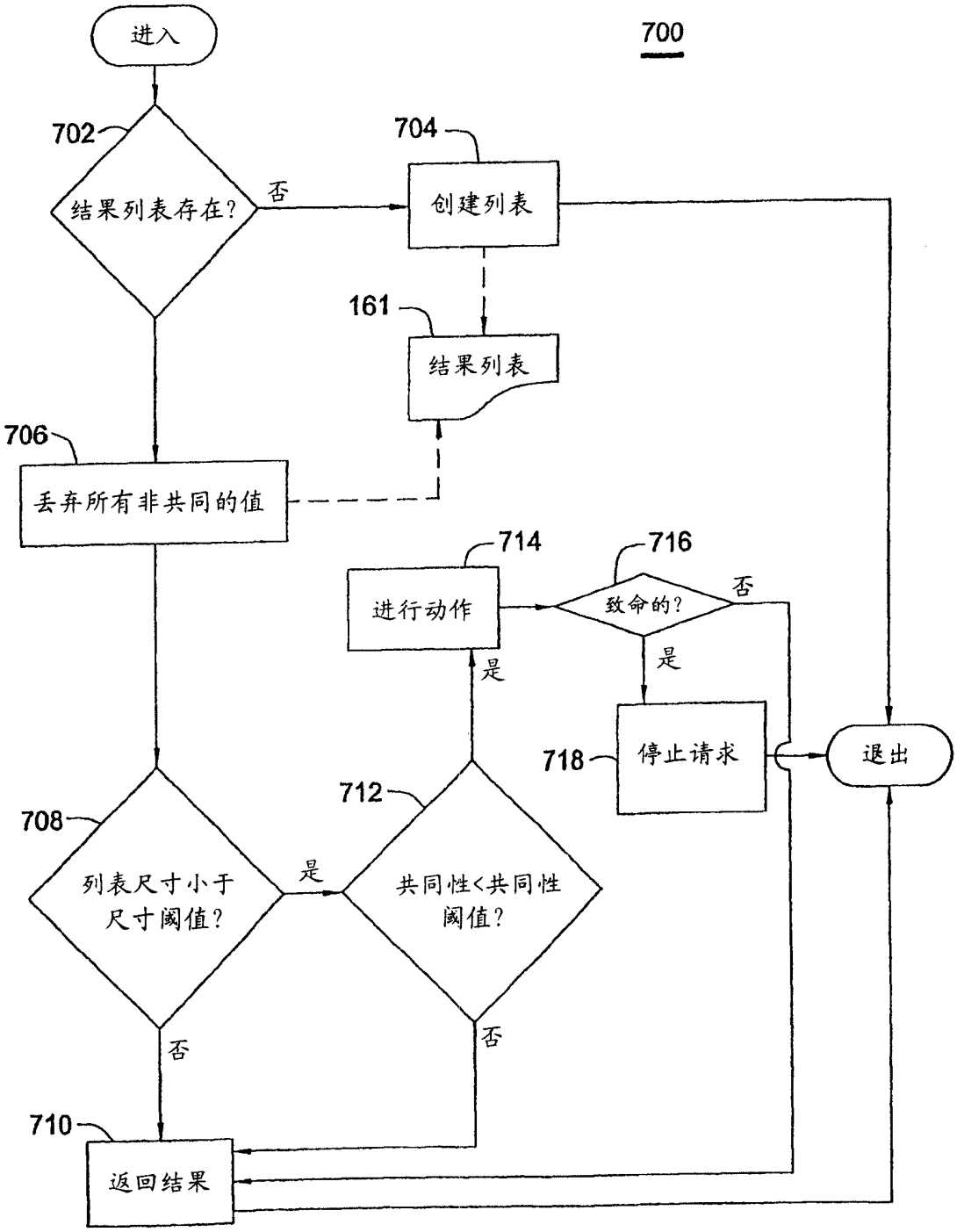

进入
700
702
结果列表存在?
否
704
创建列表
161
结果列表
706
丢弃所有非共同的值
714
进行动作
716
致命的?
否
是
是
718
停止请求
退出
712
共同性<共同性阈值?
708
列表尺寸小于尺寸阈值?
是
否
否
710
返回结果

图 7

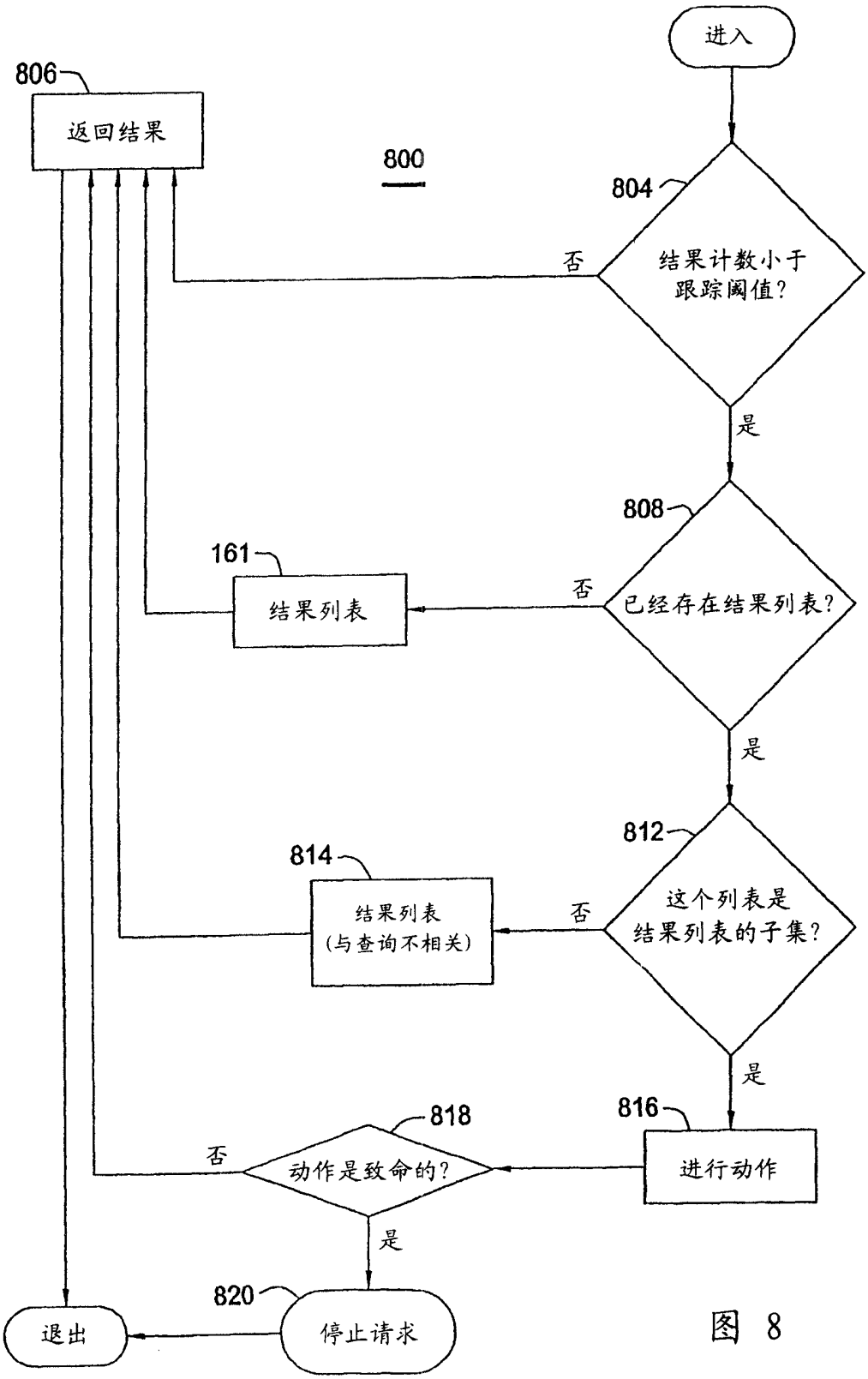

进入
806
返回结果
800
804
结果计数小于
跟踪阈值?
否
是
808
161
结果列表
否
已经存在结果列表?
是
812
814
结果列表
(与查询不相关)
否
这个列表是
结果列表的子集?
是
816
818
否
动作是致命的?
进行动作
是
820
退出
停止请求

图 8