[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)

[12] 发明专利申请公开说明书

[21] 申请号 200480019860.4

[43] 公开日 2006 年 8 月 23 日

[11] 公开号 CN 1823335A

[22] 申请日 2004.7.1
[21] 申请号 200480019860.4
[30] 优先权
[32] 2003.7.11 [33] US [31] 10/618,409
[86] 国际申请 PCT/US2004/021192 2004.7.1
[87] 国际公布 WO2005/008369 英 2005.1.27
[85] 进入国家阶段日期 2006.1.10
[71] 申请人 国际商业机器公司
地址 美国纽约
[72] 发明人 理查德·D.·得汀格
卡勒·T.·拉斯
理查德·J.·史蒂文斯
[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 付建军

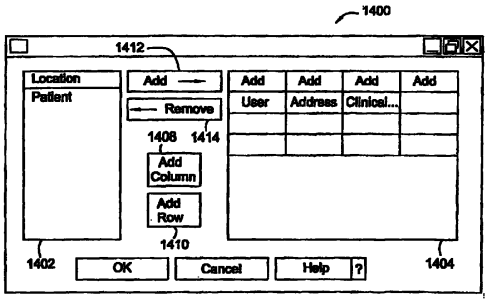
权利要求书 5 页 说明书 27 页 附图 16 页

[54] 发明名称

抽象数据链接和联接接口

[57] 摘要

一种方法、设备和产品提供了用户界面，该界面允许用户选择和排列在一个实施例中从逻辑模型所选择的结果字段。在一个实施例中，用户界面是图形用户界面(1400)并包括用于用户选择的逻辑结果字段的输入单元(1402)。单元被排列以定义逻辑结果字段(1402)之间的几何关系，该关系决定了组合语句类型，通过该类型，在查询中将单元中的逻辑结果字段(1402)。



1. 一种用于提供物理数据实体的物理字段的逻辑表示以方便查询所述物理字段的方法，该方法包括:

提供逻辑模型以在逻辑上描述所述物理字段，所述逻辑模型包括对应于相应的物理字段的逻辑字段；以及

提供运行时组件，该组件被配置为，将抽象查询转换为包含至少一个组合语句的可执行的查询，所述抽象查询包括条件和从逻辑模型的逻辑字段中选择的至少两个结果字段，每一个结果字段都在可执行的查询的组合语句中具有可执行的对应部分。

2. 根据权利要求 1 所述的方法，其中，组合语句是 UNION 语句，其中，通过使运行时组件产生 UNION 语句的 UNION 信息将至少两个结果字段相关联。

3. 根据权利要求 1 所述的方法，其中，抽象查询是用户定义的。

4. 根据权利要求 1 所述的方法，其中，可执行的查询是 SQL 语句。

5. 根据权利要求 1 所述的方法，其中，可执行的查询是 XQuery 语句。

6. 根据权利要求 1 所述的方法，其中，物理数据实体包括数据库中的多个表。

7. 根据权利要求 1 所述的方法，进一步包括提供图形用户界面，其中，在图形用户界面中指定至少两个结果字段。

8. 一种用于提供物理数据实体的物理字段的逻辑表示以方便查询所述物理字段的方法，该方法包括:

提供逻辑模型以在逻辑上描述所述物理字段，所述逻辑模型包括对应于相应的物理字段的逻辑字段;

接收相对于包括对应于相应的物理字段的逻辑字段的逻辑模型而定义的抽象查询，所述抽象查询包括条件和从逻辑模型的逻辑字段

中选择的至少两个结果字段；以及

将抽象查询转换为包含至少一个组合语句的可执行的查询，所述抽象查询包括条件和从逻辑模型的逻辑字段中选择的至少两个结果字段，每一个结果字段都在可执行的查询的组合语句中具有可执行的对应部分。

9. 根据权利要求 **8** 所述的方法，其中，物理数据实体包括数据库中的多个表。

10. 根据权利要求 **8** 所述的方法，进一步包括提供图形用户界面，其中，在图形用户界面中指定至少两个结果字段。

11. 一种用于允许构建查询的方法，包括：

提供一种图形用户界面，该界面允许用户选择和排列从在逻辑上定义数据的逻辑模型中选择的逻辑结果字段，其中，用户选择的逻辑结果字段之间的预先确定的相对几何排列定义用户选择的逻辑结果字段之间的组合关系。

12. 根据权利要求 **11** 所述的方法，其中，预先确定的相对几何排列包括用户选择的逻辑结果字段的垂直排列。

13. 根据权利要求 **11** 所述的方法，其中，组合关系被表达为包含用户选择的逻辑结果字段的表示的查询中的 **UNION** 语句。

14. 一种用于允许构建查询的方法，包括：

提供一种图形用户界面，该界面允许用户选择和排列从在逻辑上定义数据的逻辑模型中选择的逻辑结果字段，其中，用户选择的逻辑结果字段之间的第一预先确定的相对几何排列定义用户选择的逻辑结果字段之间的第一种组合关系，其中，用户选择的逻辑结果字段之间的第二预先确定的相对几何关系定义用户选择的逻辑结果字段之间的第二种组合关系。

15. 根据权利要求 **14** 所述的方法，其中，第一预先确定的相对几何排列包括选择的逻辑结果字段的垂直排列，第二预先确定的相对几何排列包括选择的逻辑结果字段的水平排列。

16. 根据权利要求 **14** 所述的方法，其中，第一种组合关系是

JOIN，第二种组合关系是 UNION。

17. 一种用于允许构建查询的方法，包括:

提供一种图形用户界面，该界面允许用户选择和排列从在逻辑上定义数据的逻辑模型中选择的逻辑结果字段，该图形用户界面包括表，该表包括多个单元，其中，相邻的单元中的用户选择的逻辑结果字段之间的预先确定的相对几何排列定义用户选择的逻辑结果字段之间的组合关系，所述组合关系是从中至少两种不同类型的组合关系中选择的。

18. 根据权利要求 17 所述的方法，其中，预先确定的相对几何排列包括用户选择的逻辑结果字段的垂直排列。

19. 根据权利要求 17 所述的方法，其中，组合关系是 UNION。

20. 根据权利要求 17 所述的方法，其中，组合关系是从 UNION 和 JOIN 中选择的。

21. 一种用于构建查询的方法，包括:

提供逻辑模型以在逻辑上描述所述物理字段，所述逻辑模型包括对应于相应的物理字段的逻辑字段;

提供一种图形用户界面，该界面允许用户选择和排列从逻辑模型中选择的逻辑结果字段;

接收用户在图形用户界面中指定第一逻辑结果字段的选择和位置的输入;

接收用户在图形用户界面中指定第二逻辑结果字段的选择和位置的输入，其中，第一和第二逻辑结果字段具有相对几何关系并定义抽象查询的至少一部分; 以及

将抽象查询转换为包含至少一个组合语句并作为相对几何关系的结果生成的可执行的查询，所述组合语句中包含第一和第二逻辑结果字段的表示。

22. 根据权利要求 21 所述的方法，其中，组合语句是 UNION。

23. 根据权利要求 21 所述的方法，进一步包括在图形用户界面中显示逻辑模型的每一个逻辑字段作为可选择的逻辑结果字段。

24. 包含图形用户界面程序的计算机可读取的介质，该程序在执行时，执行用于构建相对于包括映射到数据的物理实体的物理字段的多个逻辑字段定义的逻辑模型而定义的抽象查询的操作，该操作包括:

接收用户在图形用户界面中指定第一逻辑结果字段的选择和位置的输入；其中，图形用户界面允许用户从逻辑模型中选择逻辑结果字段，并支持用户选择的逻辑结果字段之间的组合关系；以及

接收用户在图形用户界面中指定第二逻辑结果字段的选择和位置的输入，其中，第一和第二逻辑结果字段定义抽象查询的至少一部分，所述抽象查询被转换成包含至少一个组合语句的可执行的查询，所述组合语句包含第一和第二逻辑结果字段的对应部分。

25. 根据权利要求 **24** 所述的方法，其中，组合语句是 **UNION**。

26. 包含程序的计算机可读取的介质，该程序在执行时，执行用于构建相对于包括映射到数据的物理实体的物理字段的多个逻辑字段定义的逻辑模型而定义的抽象查询的操作，该操作包括:

接收用户在图形用户界面中指定第一逻辑结果字段的选择和位置的输入；其中，图形用户界面允许用户选择和排列从逻辑模型中选择的逻辑结果字段;

接收用户在图形用户界面中指定第二逻辑结果字段的选择和位置的输入，其中，第一和第二逻辑结果字段具有相对几何关系并定义抽象查询的至少一部分；以及

将抽象查询转换为包含至少一个组合语句并作为相对几何关系的结果生成的可执行的查询，所述组合语句包含第一和第二逻辑结果字段的对应部分。

27. 根据权利要求 **26** 所述的计算机可读取的介质，其中，组合语句是 **UNION**。

28. 根据权利要求 **26** 所述的计算机可读取的介质，其中，相对几何关系是垂直。

29. 一种计算机系统，包括存储器和至少一个处理器，并进一

步包括:

一种逻辑模型，该逻辑模型包括映射到数据的物理实体的物理字段的多个逻辑字段定义，从而，逻辑模型提供了数据的逻辑视图；以及

允许用户选择和排列从逻辑模型中选择的逻辑结果字段的图形用户界面；其中，图形用户界面包括用于用户选择的逻辑结果字段的输入单元，其中，单元之间的预先定义的几何关系指定单元中的用户选择的逻辑结果字段是通过第一种组合语句类型还是通过第二种组合语句类型来相关联的。

30. 根据权利要求 **29** 所述的系统，其中，第一种组合语句类型是 **UNION**，第二种组合语句类型是 **JOIN**。

31. 根据权利要求 **29** 所述的系统，其中，预先定义的几何关系是垂直。

32. 根据权利要求 **29** 所述的系统，其中，水平相邻单元中的用户选择的逻辑结果字段被 **JOIN**。

33. 根据权利要求 **29** 所述的系统，进一步包括包含数据的物理实体的关系数据库。

抽象数据链接和联接接口

技术领域

一般而言，本发明涉及数据处理，具体来说，涉及通过逻辑框架访问数据。

背景技术

数据库是计算机化的信息存储和检索系统。最普通的数据库类型是关系数据库，这是一种其中如此定义数据以便可以以许多不同的方式重新组织和访问它的表格数据库。关系数据库管理系统 (DBMS) 是使用用于存储和检索数据的关系技术的数据库管理系统。

不管特定体系结构如何，在 DBMS 中，请求性实体（例如，应用程序、操作系统或用户）通过发出数据库访问请求来访问指定的数据库。这样的请求可以包括，例如，操作以读取、改变记录和在数据库中添加指定的记录的简单目录查询请求或事务和事务的组合。使用诸如结构化查询语言 (SQL) 之类的高级别的查询语言来作出这些请求。具体来说，使用 SQL 来作出交互式查询，以便从诸如 IBM 公司的 DB2、微软的 SQL Server 和来自 Oracle、Sybase 和 Computer Associates 的数据库产品获取信息并更新这些数据库。术语“查询”表示一组用于从存储的数据库中检索数据的命令。一般而言，查询呈现让程序员和程序选择、插入、更新、查找数据等等的位置的命令语言的形式。

查询可能常常需要对着多个表运行，以返回所需要的数据。当数据驻留在多个数据库（即，位于多个数据库服务器上）时，就是这种情况。例如，病人的记录（诊断、治疗等等）可以存储在一个数据库中，而涉及用于治疗该病人的药物的临床试验信息可以存储在另一个数据库中。因此，要访问数据，可以生成指向这些不同数据库中的每一个数据库的联合查询。如这里所使用的，术语“联合查询”一般是指

需要组合针对不同的数据库运行的查询的结果的任何查询。用来完成此任务的操作这里被称为“组合语句”。举例来说，组合语句包括诸如 JOIN 语句(包括 INNER、OUTER、LEFT 和 RIGHT)系列，UNION 语句系列之类的语句。JOIN 将来自两个不同的表或来自同一个表的数据配对。UNION 可以对两个相同的表进行操作或可以用来连接来自相似字段的数据。例如，名为 ADDRESS 的字段可以与名为 LOCATION 的字段 UNION 起来。这些字段不相同，可以，也可以不必来自于同一个表，但它们包含类似的信息。作为说明，表 I 和 II 显示了其中表被分别 JOIN 和 UNION 的 SQL 语句的示例。

TABLE I

SELECT T1.ADDRESS, T1.CITY, T1.STATE, T1.ZIP, T2.LOCATION FROM ADDRESSES T1, NEIGHBORHOOD T2 WHERE T1.ADDRESSID = T2.ADDRESSID AND T1.ZIP=55901

TABLE II

SELECT ADDRESS, CITY, STATE, ZIP FROM ADDRESSES WHERE ZIP=55901 UNION

SELECT LOCATION, NULL, NULL, NULL FROM NEIGHBORHOOD WHERE ZIP=55901

UNION 示例（表 II）将位置附加到 ADDRESS 之后，JOIN（表 I）示例将位置与它属于的地址链接。

值得注意的是，JOIN 和 UNION 只是组合语句的两个示例。组合语句的另一个示例是 ACCEPT 语句系列。此外，还存在对应的语句，或者也可以为 SQL 之外的查询语言（例如，XQuery）开发对应的语句。

通常，用户需要创建查询语句，理解基础物理数据，以便应用适当的组合语句，从而返回所希望的结果。然而，这会给用户带来很大的负担，只有具有足够的专门知识的用户才能访问。

因此，所需要的是具有灵活性的构建查询接口，具体来说，其数据可能需要链接起来或组合起来，以返回所希望的结果。

发明内容

本发明提供了使用诸如 JOIN 和 UNION 之类的组合语句访问抽象描述的物理数据的方法、系统和产品。在一个实施例中，数据通过抽象模型来进行定义，该模型包括描述和定义多个逻辑字段的元数据。提供了用于创建抽象查询的用户界面工具。该工具的一个方面给用户提供了映射必须通过第一种语句类型（例如，UNION）组合的那些字段（列）和必须通过第二种语句类型（例如，JOIN）组合的那些字段的装置。

在一个实施例中，一种方法提供了物理数据实体的物理字段的逻辑表示以方便查询所述物理字段。该方法包括提供逻辑模型以在逻辑上描述所述物理字段，所述逻辑模型包括对应于相应的物理字段的逻辑字段；并提供运行时组件，该组件被配置为，将抽象查询转换为包含至少一个组合语句的可执行的查询，所述抽象查询包括条件和从逻辑模型的逻辑字段中选择的至少两个结果字段，每一个结果字段都在可执行的查询的至少一个组合语句中具有可执行的对应部分。

另一种方法提供了物理数据实体的物理字段的逻辑表示以方便查询所述物理字段。该方法包括提供逻辑模型以在逻辑上描述所述物理字段，所述逻辑模型包括对应于相应的物理字段的逻辑字段；接收相对于包括对应于相应的物理字段的逻辑字段的逻辑模型定义的抽象查询，所述抽象查询包括条件和从逻辑模型的逻辑字段中选择的至少两个组合结果字段；并将抽象查询转换为包含至少一个组合语句的可执行的查询，所述抽象查询包括条件和从逻辑模型的逻辑字段中选择的至少两个组合结果字段，每一个结果字段都在可执行的查询的组合语句中具有可执行的对应部分。

在另一个实施例中，一种方法允许构建查询。该方法包括提供一种图形用户界面，该界面允许用户选择和排列从在逻辑上定义数据的逻辑模型中选择的逻辑结果字段，其中，用户选择的逻辑结果字段之间的预先确定的相对几何排列定义用户选择的逻辑结果字段之间的组合关系。

在另一个实施例中，一种允许构建查询的方法包括提供一种图形用户界面，该界面允许用户选择和排列从在逻辑上定义数据的逻辑模型中选择的逻辑结果字段，其中，用户选择的逻辑结果字段之间的第一预先确定的相对几何排列定义用户选择的逻辑结果字段之间的第一种组合关系，其中，用户选择的逻辑结果字段之间的第二预先确定的相对几何关系定义用户选择的逻辑结果字段之间的第二种组合关系。

在另一个实施例中，一种允许构建查询的方法包括提供一种图形用户界面，该界面允许用户选择和排列从在逻辑上定义数据的逻辑模型中选择的逻辑结果字段，该图形用户界面包括表，该表包括多个单元，其中，相邻的单元中的用户选择的逻辑结果字段之间的预先确定的相对几何排列定义用户选择的逻辑结果字段之间的组合关系。

在另一个实施例中，一种用于构建查询的方法包括提供逻辑模型以在逻辑上描述所述物理字段，所述逻辑模型包括对应于相应的物理字段的逻辑字段；提供一种图形用户界面，该界面允许用户选择和排列从逻辑模型中选择的逻辑结果字段；接收用户在图形用户界面中指定第一逻辑结果字段的选择和位置的输入；接收用户在图形用户界面中指定第二逻辑结果字段的选择和位置的输入，其中，第一和第二逻辑结果字段具有相对几何关系并定义抽象查询的至少一部分；并将抽象查询转换为包含至少一个组合语句并作为相对几何关系的结果生成的可执行的查询，所述组合语句中包含第一和第二逻辑结果字段的表示。

再一个实施例提供了包含图形用户界面程序的计算机可读取的介质，该程序在执行时，执行用于构建相对于包括映射到数据的物理实体的物理字段的多个逻辑字段定义的逻辑模型而定义的抽象查询的操作。该操作包括接收用户在图形用户界面中指定第一逻辑结果字段的选择和位置的输入；其中，图形用户界面允许用户从逻辑模型中选择逻辑结果字段，并支持用户选择的逻辑结果字段之间的关系；并接收用户在图形用户界面中指定第二逻辑结果字段的选择和位置的输入，其中，第一和第二逻辑结果字段定义抽象查询的至少一部分，所

述抽象查询被转换成包含至少一个组合语句的可执行的查询，其中包含第一和第二逻辑结果字段的对应部分。

再一个实施例提供了包含程序的计算机可读取的介质，该程序在执行时，执行用于构建相对于包括映射到数据的物理实体的物理字段的多个逻辑字段定义的逻辑模型而定义的抽象查询的操作。该操作包括接收用户在图形用户界面中指定第一逻辑结果字段的选择和位置的输入；其中，图形用户界面允许用户选择和排列从逻辑模型中选择的逻辑结果字段；接收用户在图形用户界面中指定第二逻辑结果字段的选择和位置的输入，其中，第一和第二逻辑结果字段具有相对几何关系并定义抽象查询的至少一部分；并将抽象查询转换为包含至少一个组合语句并作为相对几何关系的结果生成的可执行的查询，所述组合语句包含第一和第二逻辑结果字段的对应部分。

再一个实施例提供了计算机系统，包括存储器和至少一个处理器，并进一步包括逻辑模型，该逻辑模型包括映射到数据的物理实体的物理字段的多个逻辑字段定义，从而，逻辑模型提供了数据的逻辑视图；允许用户选择和排列从逻辑模型中选择的逻辑结果字段的图形用户界面；

其中，图形用户界面包括用于用户选择的逻辑结果字段的输入单元，其中，单元之间的预先定义的几何关系指定单元中的用户选择的逻辑结果字段是根据两种或更多种组合语句类型中的哪一种类型相关联的。

附图说明

通过参考附图中所显示的实施例，可以获得实现本发明的上文列举的特点、优点和目标并可以对它们进行详细理解的方式，上文简要概括的本发明的比较具体的说明。

然而，值得注意的是，所附的图形只显示了本发明的典型的实施例，因此，不视为对其范围进行限制，对于本发明，可以允许其他同样有效的实施例。

图 1 是说明性的计算机体系结构的方框图；

图 2 是被配置为通过物理数据源的抽象表示形式来对物理数据源处理查询的本发明的一个实施例的软件组件的关系视图；

图 3 是说明运行时组件的操作的流程图；

图 4 是说明运行时组件的操作的流程图；

图 5 是用于启动向查询添加条件的过程的图形用户界面屏幕；

图 6 是用于作为条件将出生日期添加到查询中的图形用户界面屏幕；

图 7 是向查询显示现有条件并且用户可以从中将附加条件添加到查询并执行查询的图形用户界面屏幕；

图 8 是用附加条件更新之后的图 4 的图形用户界面屏幕；

图 9 是用户可以从中将条件组合起来以构成复合条件的图形用户界面屏幕；

图 10 是已经被更新以反映组合的条件之后的图 4 的图形用户界面屏幕；

图 11 是用户可以从中将复合条件的条件取消组合的图形用户界面屏幕；

图 12 是用户可以从中选择结果字段以包括在查询中的图形用户界面屏幕；

图 13-15 是其中用户 UNION 和/或 JOIN 了所选择的结果字段的图形用户界面屏幕；

图 16 是用结果字段选择更新之后图 11 的图形用户界面屏幕；以及

图 17 是说明产生包括 UNION 语句的抽象查询的流程图；

具体实施方式

引言

本发明的一个实施例是与计算机系统一起使用的程序产品来实现的，下面将对其进行描述。程序产品的程序定义实施例（包括这里所描述的方法）的功能，并可以包含在各种承载信号的介质中。说明

性的承载信号的介质包括，但不仅限于：(i) 永久存储在非可写入的存储介质（例如，可由 CD-ROM 驱动器读取的诸如 CD-ROM 光盘之类的计算机内的只读存储器设备）中的信息；(ii) 存储在可写入存储介质（例如，软盘驱动器内的软盘或硬盘驱动器）上的可变的信息；或 (iii) 诸如通过计算机或电话网络，包括无线通信，通过通信介质传输到计算机的信息。后一实施例特别包括从因特网和其他网络下载的信息。这样的承载的信号介质，当携带实现本发明的功能的计算机可读的指令时，表示本发明的实施例。

一般而言，被执行以实现本发明的实施例的例程，可以是操作系统的一部分或特定的应用程序、组件、程序、模块、对象，或指令序列。本发明的软件通常包括将被本地计算机转换为机器可读的格式并因此转换为可执行的指令的多个指令。此外，程序还包括驻留在程序本地或被发现在存储器或者存储设备中的变量和数据结构。此外，可以基于应用程序（在本发明的特定实施例中实现了程序）识别下文所描述的各种程序。然而，应该理解，随后的任何特定的术语都只是为了方便，如此本发明不应该只限于在这样的术语所标识和/或暗指的任何特定的应用程序中。

本发明的实施例用于构建查询。在一个实施例中，用户指定条件（例如，如在 SQL 查询的 WHERE 子句中）以及结果标准（例如，如在 SELECT 子句中）。组合语句根据搜索到的结果将结果标准的组件关联。例如，组合语句包括 JOIN、UNION、ACCEPT 和它们相应的系列成员（例如，在 JOIN 的情况下，为 INNER、OUTER、LEFT 和 RIGHT）。为了方便起见并为了说明，将强调 JOIN 和 UNION。然而，应该理解，本发明不仅限于特定类型的组合语句，本发明也不涉及特定查询语言。如此，对 SQL 的引用只是为了说明，而不对本发明作出限制，同样地适用于已知或未知的任何查询语言。

在一个实施例中，提供了特定数据定义框架（这里也简称为数据抽象模型 (DAM)），用于查询数据，与物理上表示数据的特定方式无关。DAM 包括描述和定义映射到物理数据的多个逻辑字段的元数据。

然而，虽然是就构建并相对于逻辑模型来执行的查询来描述本发明的实施例的，但是本发明不仅限于任何任何特定的逻辑模型。相应地，这里所说明的实施例只是说明性的。

环境的物理视图

图 1 描述了其中可以实现本发明的实施例的网络系统 100 的方框图。一般而言，网络系统 100 包括客户端（即，一般而言，诸如用户或应用程序之类的任何请求性实体）计算机 102（显示了三个这样的客户端计算机）和至少一个服务器计算机 104（显示了一个这样的服务器计算机）。客户端计算机 102 和服务器计算机 104 通过网络 126 连接在一起。一般而言，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。然而，值得注意的是，本发明的各个方面不必在分布式环境中实现。如此，一般而言，客户端计算机 102 和服务器计算机 104 分别代表发出查询的任何请求性实体（如用户或应用程序）和被配置为处理查询的接收性实体。

客户端计算机 102 包括通过总线 130 连接到存储器 112、存储器 114、输入设备 116、输出设备 119 和网络接口设备 118 的中央处理单元 (CPU) 110。输入设备 116 可以是向客户端计算机 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、跟踪球、或语音识别单元、音频/视频播放器等等。输出设备 119 可以是向用户提供输出的任何设备，例如，任何常规显示屏幕。虽然所显示的是与输入设备 116 分开的，但是，输出设备 119 和输入设备 116 可以组合起来。例如，可以使用具有集成的触摸屏的显示屏幕，具有集成的键盘的显示器或与文字语言转换器结合的语音识别单元。

网络接口设备 118 可以是被配置为允许通过网络 126 在客户端计算机 102 和服务器计算机 104 之间进行网络通信的任何进入/退出设备。例如，网络接口设备 118 可以是网络适配器或其他网络接口卡 (NIC)。

优选情况下，存储器 114 是直接访问存储器 (DASD)。虽然显

示为单个单元，但是，它可以是固定和/或可移动存储设备的组合，如固定磁盘驱动器、软盘驱动器、磁带驱动器、可移动存储卡或光存储器。存储器 112 和存储器 114 可以是跨多个主要和辅助存储设备的一个虚拟地址空间的一部分。

优选情况下，存储器 112 是充分大，可以容纳本发明的所需要的编程和数据结构的随机存取存储器。尽管存储器 112 被显示为单个实体，但是，应该理解，事实上，存储器 112 可以包括多个模块，存储器 112 可以在多个级别存在，从高速寄存器和高速缓存到较低速度的但较大的 DAMM 芯片。

为便于说明，存储器 112 包含操作系统 124。可以优先使用的说明性的操作系统包括 Linux 和微软的 Windows®。一般而言，可以使用支持这里所说明的功能的任何操作系统。

存储器 112 还包含浏览器程序 122，当在 CPU 110 上执行该浏览器程序时，支持在各种服务器 104 之间浏览并将网络地址定位到一个或多个服务器 104 上。在一个实施例中，浏览器程序 122 包括基于 Web 的图形用户界面 (GUI)，该界面允许用户显示超文本标记语言 (HTML) 信息。然而，一般而言，浏览器程序 122 可以是能够呈现从服务器计算机 104 传输的信息的任何基于 GUI 的程序。

服务器计算机 104 在物理上可以以类似于客户端计算机 102 的方式进行配置。相应地，一般而言，显示的服务器计算机 104 包括 CPU 130、存储器 132，以及存储设备 134，它们通过总线 136 彼此连接在一起。存储器 132 可以是充分大的随机存取存储器，可以容纳位于服务器计算机 104 上的所需要的编程和数据结构。

一般而言，服务器计算机 104 在驻留在存储器 132 中的操作系统 138 的控制下。操作系统 138 的示例包括 IBM OS/400®, UNIX, Microsoft Windows® 等等。一般而言，可以使用能够支持这里所描述的功能的任何操作系统。

存储器 132 进一步包括一个或多个应用程序 140 和抽象查询界面 146。应用程序 140 和抽象查询界面 146 是软件产品，包括在

不同的时间驻留在计算机系统 100 中的各种存储器和存储设备中的多个指令。当被服务器 104 中的一个或多个处理器 130 读取和执行时，应用程序 140 和抽象查询界面 146 使计算机系统 100 执行那些实现本发明各个方面所需要的步骤或元素。应用程序 140（一般而言，任何请求性实体，包括操作系统 138，在最高级别，用户）针对数据库发出查询。可以向其发出查询的说明性的源包括本地数据库 $156_1...156_N$，远程数据库 $157_1...157_N$，统称为数据库 156-157。为便于说明，所显示的数据库 156 为存储器 134 中的数据库管理系统 (DBMS) 154 的一部分。一般而言，如这里所使用的，术语“数据库”是指任何数据集合，不管特定的物理表示形式。作为说明，可以根据关系架构（可通过 SQL 查询来进行访问）或根据 XML 架构（可通过 XML 查询来进行访问）来组织数据库 156-157。然而，本发明不仅限于特定架构，并可以扩展到目前未知的架构。如这里所使用的，术语“架构”一般是指通过数据知识库抽象 148 描述的数据的特定布局。

在一个实施例中，根据每一个应用程序 140 中所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 发出的查询可以是预先定义的（即，作为应用程序 140 的一部分被硬编码），也可以响应输入（例如，用户输入）来生成。不论是哪一种情况，都可以使用抽象查询界面 146 所定义的逻辑字段来构成查询（这里简称为“抽象查询”）。具体来说，在抽象查询中所使用的逻辑字段是通过抽象查询界面 146 的数据抽象模型 (DAM) 148 来进行定义的。抽象查询被运行时组件 150 处理，该运行时组件将抽象查询转换为与一个或多个数据库 156-157 中所包含的数据的物理表示形式一致的形式（这里简称为“具体查询”）。具体来说，此处理是由运行时组件 150 的物理查询编制器 161 来执行的。运行时组件 150 还包括分析工具 162，这样叫，是因为它能实现这里所描述的数据分析功能。为便于说明，分析工具 162 包括 DAM 生成器 164、查询增强器 166 和表编制器 168。值得注意的是，运行时组件 150 的功

能只是说明性的。那些精通本技术的人员将认识到，可以在别处（例如，在数据库管理系统 154 本身中）实现这些功能。下面将进一步描述抽象查询界面 146 的每一个组件/功能。

由运行时组件 150 处理的抽象查询可以被配置为访问数据和返回结果，或修改（即，插入、删除或更新）数据。在一个实施例中，由用户通过图形用户界面 (GUI) 指定查询的元素。由应用程序 140，具体来说，由图形用户界面 (GUI) 编制器 144 来生成 GUI 的内容。在特定实施例中，GUI 内容是超文本标记语言 (HTML) 内容，可以用浏览器程序 122 呈现在客户端计算机系统 102 上。相应地，存储器 132 包括超文本传输协议 (http) 服务器进程 138（例如，Web 服务器），用于服务于来自客户端计算机 102 的请求。例如，进程 138 可以响应访问说明性地驻留在服务器 104 上的数据库 156 的请求。传入的客户端的对数据库 156-157 中的数据的请求调用应用程序 140。当由处理器 130 执行时，应用程序 140 使服务器计算机 104 执行实现本发明的各个方面的步骤或元素，包括访问数据库 156-157。在一个实施例中，应用程序 140 包括多个被配置为构建 GUI 元素（然后，它们由浏览器程序 122 呈现）的多个 servlet。在通过应用程序 140 访问远程数据库 157 的情况下，给数据抽象模型 148 配置了标识包含要被检索的数据的数据库的位置规范。下面将比较详细地描述此后一实施例。

图 1 只是网络客户端计算机 102 和服务器计算机 104 的一个硬件/软件配置。本发明的实施例可以应用于任何可比较的硬件配置，不管计算机系统是复杂的多用户计算设备，单用户工作站，还是没有它们自己的非易失性存储器的网络设备。此外，应该理解，尽管引用了特定标记语言，包括 HTML，但是，本发明不仅限于特定语言、标准或版本。相应地，那些精通本技术的人员将认识到，本发明可以适用于其他标记语言以及非标记语言，本发明还可以适用于特定标记语言中的未来的变化以及目前未知的其他语言。同样，图 1 所示的 http 服务器进程 138 只是说明性的，可以支持任何已知和未知的协议的其

他实施例也是可以的。

环境的逻辑/运行时视图

图 2A-B 显示了本发明的多个相互关联的组件。请求性实体（例如，其中一个应用程序 140）发出如请求性实体的相应的应用程序查询规范 142 所定义的查询 202。所产生的查询 202 这里一般是指“抽象查询”，因为根据抽象（即，逻辑）字段而不是通过对数据库 156-157 中的基础物理数据实体的直接引用来构成。结果，可以定义与所使用的特定基础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 可以包括用于进行数据选择的标准（选择标准 204）和基于选择标准 204 的要被返回的字段的显式规范（返回数据规范 206）。

由应用程序查询规范 142 指定的并用于构成抽象查询 202 的逻辑字段是由数据抽象模型 148 定义的。一般而言，数据抽象模型 148 作为一组逻辑字段暴露信息，这些逻辑字段可以在应用程序 140 发出的查询（例如，抽象查询 202）内使用，以指定进行数据选择的标准，并指定从查询操作返回的结果数据的形式。逻辑字段是独立于在数据库 156-157 中使用的基础数据表示而定义的，从而允许形成松散耦合到基础数据表示的查询。DAM 148 的逻辑字段被映射到其中的数据可以位于单个数据的知识库（即，源）或多个不同的数据知识库中。如此，DAM 148 可以提供一个或多个基础数据知识库的逻辑视图。通过使用数据知识库的抽象表示形式，可以比较轻松地改变或替换基础物理表示形式，而不会影响作出更改的应用程序。相反，抽象表示形式可以变化，而没有应用程序所需的更改。此外，可以定义多个抽象数据表示形式，以支持针对可以具有不同的默认值或所需的字段的相同基础数据库架构的不同的应用程序。

一般而言，数据抽象模型 148 包括多个字段规范 208_1、208_2、208_3、208_4 和 208_5（作为示例，显示了五个），统称为“字段规范 208”。具体来说，为可用于构成抽象查询的每一个逻辑字段，提供了字段规范。每一个字段规范包括逻辑字段名称 210_1、210_2、210_3、210_4、210_5

（统称为，字段名称 210）和关联的访问方法 212_1、212_2、212_3、212_4、212_5（统称为，访问方法 212）。访问方法根据这里被称为物理位置参数的参数将逻辑字段名称关联（即，映射）到数据库（例如，其中一个数据库 156）中的特定物理数据表示形式 $214_1, 214_2...214_N$。作为说明，显示了两个数据表示，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 表示，任何其他数据表示，已知的或未知的，都是可以的。

取决于要支持的不同类型的逻辑字段的数量，可以有任意数量的访问方法。在一个实施例中，提供了简单字段、过滤的字段和合成字段的访问方法。字段规范 2081、2082 和 2085 分别示范了简单字段访问方法 2121、2122 和 2125。直接将简单字段映射到基础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 所示的简单字段访问方法 212_1 将逻辑字段名称 210_1 ("FirstName") 映射到名为“contact”的表中的名为“f_name”的列，其中，表名称和列名称是访问方法 212_1 的物理位置参数。字段规范 208_3 示范了过滤的字段访问方法 212_3。过滤的字段标识关联的物理实体，并提供用于定义物理数据表示内的项目的特定子集的规则。在图 2B 中提供了一个示例，在该示例中，过滤的字段访问方法 212_3 将逻辑字段名称 210_3 (“AnytownLastName”) 映射到名为“contact”的表中的名为“I_name”的列中的物理实体，并为城市 Anytown 中的个体定义过滤器。过滤的字段的另一个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，并将数据只限制到为纽约州定义的那些邮政编码。字段规范 208_4 示范了合成字段访问方法 212_4。合成访问方法使用作为访问方法定义的一部分提供的表达式从一个或多个物理字段来计算逻辑字段。如此，可以计算出不存在于基础数据表示的信息。在图 2B 中说明的示例中，合成字段访问方法 2123 将逻辑字段名称 2103 “AgeInDecades”映射到“AgeInYears/10”。另一个示例是通过将售价字段乘以营业税率构成的营业税字段。

值得注意的是，图 2B 所示的数据抽象模型 148 只是所选择的

逻辑字段规范的说明，并不是全面的。如此，图 2B 所示的抽象查询 202 包括一些逻辑字段（在数据抽象模型 148 中没有显示规范），如“州”和“街道”。

基础数据的任何给定数据类型（例如，日期、十进制数字等等）的格式可以变化。相应地，在一个实施例中，字段规范 208 包括反映了基础数据的格式的类型属性。然而，在另一个实施例中，字段规范 208 的数据格式不同于关联的基础物理数据，在这样的情况下，访问方法负责以请求性实体采用的适当的格式返回数据。如此，访问方法必须知道采用什么格式的数据（即，根据逻辑字段）以及基础物理数据的实际格式。然后，访问方法可以将基础物理数据转换为逻辑字段的格式。

作为示例，图 2A 所示的数据抽象模型 148 的字段规范 208 是映射到以关系数据表示 214_2 表示的数据的逻辑字段的代表。然而，数据抽象模型 148 的其他实例将逻辑字段映射到诸如 XML 之类的其他物理数据表示。此外，在一个实施例中，数据抽象模型 148 被配置有过程数据表示的访问方法。

下面的表 I 中显示了对应于图 2 所示的抽象查询 202 的说明性的抽象查询。作为说明，使用 XML 来定义数据知识库抽象 148。然而，也可以使用任何其他语言。

表 I - 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>
```

为便于说明，表 I 中所显示的抽象查询包括选择规范（005-014 行），其中，包含选择标准和结果规范（015-019 行）。在一个实施例中，选择标准包括字段名称（对于逻辑字段），比较运算符（=, >, <, 等等）和值表达式（什么是与其进行比较的字段）。在一个实施例中，结果规范是将作为查询执行的结果返回的抽象字段的列表。抽象查询中的结果规范可以包括字段名称和排序标准。

下面的表 II 中显示了对应于表 I 中的抽象查询的数据抽象模型 148 的说明性实例。作为说明，使用 XML 来定义数据抽象模型 148。然而，也可以使用任何其他语言。

表 II - 数据抽象模型示例

```
001  <?xml version="1.0"?>
002  <DataRepository>
003   <Category name="Demographic">
```

```
004     <Field queryable="Yes" name="FirstName" displayable="Yes">
005         <AccessMethod>
006           <Simple columnName="f_name" tableName="contact"></Simple>
007         </AccessMethod>
008         <Type baseType="char"></Type>
009     </Field>
010     <Field queryable="Yes" name="LastName" displayable="Yes">
011         <AccessMethod>
012           <Simple columnName="l_name" tableName="contact"></Simple>
013         </AccessMethod>
014         <Type baseType="char"></Type>
015     </Field>
016     <Field queryable="Yes" name="State" displayable="Yes">
017         <AccessMethod>
018           <Simple columnName="state" tableName="contact"></Simple>
019         </AccessMethod>
020         <Type baseType="char"></Type>
021     </Field>
022   </Category>
023  </DataRepository>
```

注意，004-009 行对应于图 2B 所示的 DAM 148 的第一字段规范 2081，而 010-015 行对应于第二字段规范 2082。为了简便起见，图 2B 没有显示表 I 中定义的其他字段规范。还请注意，表 I 说明了类别，在此情况下“Demographic”。类别是一个或多个逻辑字段的组合。在本示例中，“First Name”、“Last Name”和“State”是属于共同的类别“Demographic”的逻辑字段。

在任何情况下，数据抽象模型 148 包含（或引用）将逻辑字段映射到物理数据的至少一个访问方法。然而，前述的实施例只是说明性的，逻辑字段规范可以包括各种其他元数据。在一个实施例中，访问方法进一步配置有定义与逻辑字段关联的数据的位置的位置规范。如此，数据抽象模型 148 被延长，以包括许多数据源（可以是本地的和/或跨网络环境分布）的说明。数据源可以使用多个不同的数据表示和数据访问技术。如此，提供了基础结构，该基础结构能够利用当今流行的分布式环境。标题为“REMOTE DATA ACCESS AND INTEGRATION OF DISTRIBUTED DATA SOURCES THROUGH DATA SCHEMA AND QUERY ABSTRACTION”的并授予给 IBM 公司的美国专利申请 No. 10/131,984 中比较详细地描述了访问许多数据源的一个方法。

图 3 显示了示范了运行时组件 150 的操作的一个实施例的说明性运行时方法 300。当运行时组件 150 作为输入接收到抽象查询（如图 2 所示的抽象查询 202）的实例时，在步骤 302 中输入方法 300。在步骤 304 中，运行时组件 150 读取并分析抽象查询的实例，并定位单个选择标准和所希望的结果字段。在步骤 306 中，运行时组件 150 进入处理抽象查询中存在的每一个查询选择标准语句的循环（包括步骤 306、308、310 和 312），从而构建具体查询的数据选择部分。在一个实施例中，选择标准包括字段名称（对于逻辑字段），比较运算符(=, >, <,等等)和值表达式(什么是与其进行比较的字段)。在步骤 308 中，运行时组件 150 使用来自抽象查询的选择标准的字段名称，以查找数据知识库抽象 148 中的字段的定义。如上所述，字段定义包括用于访问与字段关联的物理数据的访问方法的定义。然后，运行时组件 150 为正在被处理的逻辑字段构建（步骤 310）Concrete Query Contribution（具体查询成分）。如这里所定义的，Concrete Query Contribution 是用于基于当前逻辑字段执行数据选择的具体查询的一部分。具体查询是以诸如 SQL 和 XML Query 之类的语言表示的查询，与给定物理数据知识库（例如，关系数据库或 XML 知识库）的数据一致。相应地，具体查询用于定位并从由图 1 所示的数据库 156-157 代表的物理数据知识库中检索数据。然后，将为当前字段生成的 Concrete Query Contribution 添加到 Concrete Query Statement (具体查询语句) 。然后，方法 300 返回到步骤 306，以开始抽象查询的下一字段的处理。相应地，为抽象查询中的每一个数据选择字段，重复进入步骤 306 中的过程，从而将额外的内容提供到要执行的最终查询。

在构建具体查询的数据选择部分之后，运行时组件 150 标识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，抽象查询定义将作为查询执行的结果返回的抽象字段的列表，这里被称为“结果规范”。抽象查询中的结果规范可以包括字段名称和排序标准。相应地，方法 300 在步骤 314 中进入循环(由步骤 314、316、318 和 320

进行定义），以将结果字段定义添加到正在生成的具体查询。在步骤 316 中，运行时组件 150 在数据知识库抽象 148 中查找结果字段名称（从抽象查询的结果规范中），然后从数据知识库抽象 148 中检索结果字段定义，以标识要为当前逻辑结果字段返回的数据的物理位置。然后，运行时组件 150 为逻辑结果字段构建（作为步骤 318）（标识要返回的数据的物理位置的具体查询的） Concrete Query Contribution。然后，在步骤 320 中，将 Concrete Query Contribution 添加到 Concrete Query Statement。一旦处理完成抽象查询中的每一个结果规范，就可以在步骤 322 中执行查询。

参考图 4 描述根据步骤 310 和 318 的为逻辑字段构建 Concrete Query Contribution 的方法 400 的一个实施例。在步骤 402 中，方法 400 查询与当前逻辑字段关联的访问方法是否为简单访问方法。如果是，则基于物理数据位置信息构建 Concrete Query Contribution（步骤 404），然后，根据上文所描述的方法 300，继续处理过程。否则，处理过程持续到步骤 406，以查询与当前逻辑字段关联的访问方法是否为过滤访问方法。如果是，则基于某些物理数据实体的物理数据位置信息，构建 Concrete Query Contribution（步骤 408）。在步骤 410 中，用使用的额外的逻辑（过滤器选择）将 Concrete Query Contribution 扩展到与物理数据实体关联的子集数据。然后，根据上文所描述的方法 300，继续处理过程。

如果访问方法不是过滤访问方法，则处理从步骤 406 进入步骤 412，在此，方法 400 查询访问方法是否为合成访问方法。如果访问方法是合成访问方法，则在步骤 414 中定位和检索合成字段表达式中的每一个子字段介绍信的物理数据位置。在步骤 416 中，用合成字段表达式的物理字段位置信息代替合成字段表达式的逻辑字段引用，从而生成 Concrete Query Contribution。然后，根据上文所描述的方法 300，继续处理过程。

如果访问方法不是合成访问方法，则处理从步骤 412 进入步骤 418。步骤 418 是作为本发明的实施例的任何其他访问方法类型的代

表。然而，应该理解，还可以有其中不必实现全部可用的访问方法的其他实施例。例如，在特定实施例中，只使用简单访问方法。在另一个实施例中，只使用简单访问方法和过滤访问方法。

如上所述，如果逻辑字段指定不同于基础物理数据的数据格式，则可能需要执行数据转换。在一个实施例中，当根据方法 400 为逻辑字段构建 Concrete Query Contribution 时，为每一个相应的访问方法执行初始转换。例如，可以作为步骤 404、408 和 416 的一部分、或紧后面的步骤来执行转换。在步骤 322 中执行查询之后，执行随后的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基础物理数据的格式相同，则不需要进行转换。

图形用户界面

如上所述，一个实施例为用户提供了 GUI，通过该 GUI，可以合成和执行查询。GUI 屏幕（例如，由应用程序 140 的 GUI 编制器 144 构建）一般提供了搜索条件和用户可以选择的关联的值。用户的选择用于构建应用程序查询规范 142。然后，查询可以以上文所描述的方式执行。

为了便于说明，定义某些术语是有帮助的。从上面的说明可以看出，为对知识库 154 内的一组所希望的数据库记录执行搜索，请求性实体（例如，应用程序 140）通过将一个或多个“操作数”和“逻辑算子”组合以构成搜索表达式，来构建查询。操作数与算子一起标识所希望的搜索。每一个操作数都可以是为知识库 154 中的元素的参数定义值的比较操作（通过比较运算符定义，例如，>, <, =）。例如，操作数可以是表示具有等于日期值 1942/01/01 的"DateOfBirth"参数的搜索结果的请求的 "(DateOfBirth = '1942/01/01')"。另一个说明性的操作数可以是表示具有大于 1942/01/01 的 "DateOfBirth" 参数的搜索结果的请求的 "(DateOfBirth > '1942/01/01')"。可以通过表示操作数之间的逻辑关系的逻辑算子来将两个或更多操作数关联。逻辑算子是诸如逻辑 AND、OR 和 NOT 之类的逻辑连接符。每一个操作数，或具有逻辑算子的操作数都定义单个搜索条件。

为了便于说明，单一的操作数这里被称为“简单条件”或只是“条件”。通过逻辑算子相关联的两个或更多操作数/条件构成了“复合条件”或“复合条件”。简单条件或复合条件可以构成由应用程序 140 执行的搜索表达式（即，查询）的一部分。

现在请参看图 5-16，显示了一系列图形用户界面，说明通过应用程序 140 定义的用户界面。作为说明，如图 5-16 所示的图形用户界面是访问医学数据所特定的。然而，可以使用本发明的实施例，它们具有任何信息类型，其中包括，财务信息、雇用信息等等。一般而言，如图 5-16 所示的图形用户界面允许用户构建包括由用户添加的条件的查询。回想一下，如这里所定义的，简单条件是比较操作。(DateOfBirth = '1942/01/01') 是说明性的简单条件。如此，将简单条件添加到查询中一般涉及允许用户选择操作数和比较运算符（例如，>, <, =）。在一个实施例中，当用户通过浏览器程序 122 最初调用应用程序 140 时，应用程序 140（具体来说，GUI 编制器 144）返回由浏览器程序 122 以第一 GUI 600 的形式呈现的 HTML 内容，如图 5 所示。GUI 600 是用户用来将条件添加到查询中的一系列屏幕中的第一个屏幕。一般而言，GUI 600 包括多个条件类别 602-610（每一个类别都具有关联的单选按钮），用户可以从中进行选择。所显示的条件类别包括“人口资料”602、“测试和实验室结果”604、“报告”606、“使用 ICD-9 的诊断”608，以及“使用 DRG 的诊断”610。每一个条件类别都具有关联的字段 612-620，可以从中选择值/向其中输入值。一些字段（例如，字段 612-616）是下拉菜单，而另一些字段是用于接收用户输入的文本框（例如，字段 618-620）。在后一种情况下，字段可以具有关联的“浏览”按钮 622-624 以帮助用户选择有效的值。

一旦选择条件类别和值，用户点击“下一步”按钮 626。点击“下一步”按钮 626 使浏览器程序 622 呈现（根据应用程序 640 所提供的信息）继续添加条件的过程所需要下一个适当的图形用户界面。如此，可以向用户添加条件所需要的一系列图形用户界面。作为示例，假设用户从下拉菜单 612 选择了人口条件类别 602 和“出生日期”

值。在按下“下一步”按钮 626 时，向用户呈现如图 6 所示的第二 GUI 700。GUI 700 包括比较运算符下拉菜单 702（用户可以从中选择比较运算符（例如，>, <, =））和用户可以根据规定的格式（例如，"yyyy/mm/dd"）向其中输入日期的日期字段 704。当用户点击 OK 按钮 706 时，完成添加“出生日期”条件的过程。

当用户完成添加条件的过程（例如，如在点击 OK 按钮 706 之后），向用户呈现如图 7 所示的 GUI 800。在条件列 802 中显示了所产生的条件。在本示例中，条件列 802 的第一行显示了使用上文所描述的 GUI 600 和 700 添加“出生日期”条件。用户可以通过点击“添加条件”按钮 804 添加另一个条件。通过对于每一个其他可用的条件类别和值重复类似的步骤，可以将任意数量的条件添加到查询中。作为说明，如图 8 所示的刷新的/更新的 GUI 800 总共显示了三种条件（包括上文所描述的“出生日期”条件），通过一步一步地翻过用于添加条件的图形用户界面（例如，GUI 600 和 700）来添加每一个条件。此外，虽然未显示，添加条件可能需要任意数量的其他图形用户界面。具体来说，对于比较复杂的条件，如范围，可能需要多个额外的图形用户界面。根据本发明的各个方面，那些精通本技术的人员可以确定这样的图形用户界面的特定数量、内容、设计和元素。

在添加第一条件之后，通过布尔逻辑值/运算符，这里被称为第一级别的布尔逻辑值，每一个随后条件彼此关联到第一条件。请参看图 8，在一对列 902、904 中显示了第一级别布尔逻辑值。从第一列 902 中的下拉菜单 906、908 中选择第一级别的布尔逻辑值 AND 和 OR。AND/OR 下拉菜单位于每一个条件之间。相应地，在如图 8 所示的例图中，第一下拉菜单 906 位于“出生日期”条件和“性别”条件之间，第二下拉菜单 908 位于“性别”条件和“血色素”条件之间。在一个实施例中，第一级别的布尔逻辑值默认到第一条件之后添加的每一个条件的 AND。然后，用户可以使用下拉菜单将默认值改变为 OR。或者，用户可以通过选中第二列 904 中的适当的 NOT 复选框来否定条件。

一旦将两个或更多条件添加到查询，然后，可以通过布尔逻辑值（这里被称为第二级别的布尔逻辑值)将两个或更多条件组合在一起，以产生复合（或组合的）条件。((employeeName = 'Rich') OR (employeeName = 'John')) 是复合条件的示例。此外，可以通过布尔逻辑值将简单条件和复合条件连接在一起，以产生条件的层次结构。在一个实施例中，用于将条件组合在一起的第一图形元素是“组合条件”按钮 910。

在一个实施例中，按下“组合条件”按钮 910 使应用程序 140 生成如图 9 所示的 GUI 1000。GUI 1000 显示每一个可用的条件和每一个条件的关联的复选框 1002。用户通过选中适当的复选框 1002 选择哪些条件将被组合在一起。为便于说明，选择“性别”和“血色素”条件。GUI 1000 进一步提供了“AND 组”按钮 1004 和“OR 组”按钮 1006，以便对所选择的组进行 AND 或 OR 运算。作为说明，假设用户点击“AND 组”按钮 1004。通过图 10 中的更新的 GUI 800 显示了此操作的结果。具体来说，只有两个条件 1102、1104 保留，第二条件 1104 是包括通过布尔 AND 相关联的子条件的复合条件。此外，两个条件 1102、704 本身通过布尔 AND 相关联，如下拉菜单 906 中所显示的。

现在假设，用户需要取消组合第二条件 1104。为执行此任务，GUI 900（如图 10 所示）包括“撤消组合的条件”按钮 1106。按下“撤消组合的条件”按钮 1106 使应用程序 140 生成如图 11 所示的 GUI 1200。GUI 1200 显示每一个现有的条件和关联的复选框 1202。在其他实施例中，在 GUI 1200 中只显示了复合条件（在此情况下，如第二条件）。在任何情况下，为取消组合复合条件，用户选中适当的复选框 1202，然后点击 OK 按钮 1204。在此情况下，GUI 900 和图 8 显示了取消组合第二条件 1104 的结果。即，将条件返回到它们的原始未组合的状态。

用户可以在如图 8 和图 10 所示的 GUI 900 的搜索摘要部分 920 中查看给定查询的当前结构。在一个实施例中，通过点击“刷新搜

索摘要”按钮 922，更新搜索摘要部分 920 中显示的查询。

一旦定义所希望的条件，甚至在定义条件之前，用户可以定义结果字段（即，用户希望返回的并满足条件标准的字段）。在一个实施例中，用户点击“改变输出数据”按钮 1108（图 10 中所显示的）。作为响应，GUI 编制器 144 呈现图 12 的“结果字段选择 GUI 1300”。GUI 1300 包括“可用的结果字段”窗口 1302 和“选择的结果字段”窗口 1304。窗口 1302 显示用户可以从中选择的所有可用的结果字段。窗口 1304 显示用户所选择的字段。在本示例中，用户选择了“位置”、“地址”、“病人 ID”、“临床信息”和“用户”。在一个实施例中，GUI 1300 是支持拖放的，有助于选择结果字段。在作出所希望的选择之后，用户按下“下一步”按钮 1306，并向用户呈现图 13 的“结果字段链接 GUI 1400”。

“结果字段链接 GUI”1400 一般在“结果字段选择 GUI”1300 中包括由用户选择的结果字段的列表 1402。GUI 1400 还包括具有多个列 1406A-D（显示了四个）的表 1404。每一列都具有许多单元（显示了每一列有三个）。只作为说明，表 1404 包括四列，每一列都具有三个单元。然而，表 1404 的尺寸可以由用户随意地操纵。在一个实施例中，用户通过点击“添加列”按钮 1408 添加另一个列，并通过点击“添加行”按钮 1410，将另一行添加到每一列。

用户通过选择表 1404 中的可用结果字段（在列表 1402 中）的相对位置，确定所产生的查询的结构。在一个实施例中，用户通过突出显示选择的单元，然后，突出显示列表 1402 中的一个结果字段，然后点击“添加”按钮 1412，填充表 1404 的单元。通过点击“删除”按钮 1414，可以将填充的单元解除填充。在另一个实施例中，用户只需（从列表 1402 中）拖动并将所选择的结果字段放到表 1404 的单元中。为便于说明，用户用“用户、地址、临床信息”填充第一行的开头三个单元，如图 14 所示。相应地，列表 1402 中的其余结果字段是“位置”和“病人”。在一个实施例中，同一行（即，水平地排列）上的相邻单元中的字段被 JOIN，假设字段位于各自的表中。相应地，

对应于如图 14 所示的表 1404 的 SQL 语句如下:

对应的 SQL 语句

SELECT T1.USER, T1.ADDRESS, T2.CLINICAL_INFO FROM TABLE1 T1, TABLE2 T2

在一个实施例中，UNION 以垂直方式排列的字段。例如，图 15 显示了一种排列，其中，“病人”和“位置”被添加到“用户”和“地址”紧下面的下一行中的对应的单元。相应地，“病人”和“用户”被 UNION，“位置”和“地址”被 UNION。注意，“临床信息”紧下面的单元是空的。在 SQL 语句中，此空的单元将是空值。相应地，此排列的对应的 SQL 语句如下:

对应的 SQL 语句

SELECT USER, ADDRESS, CLINICAL_INFO FROM TABLE1
UNION
SELECT PATIENT AS USER, LOCATION AS ADDRESS, NULL, FROM TABLE2.

相应地，本发明的用户界面的一个实施例允许用户控制在数据被抽象地表示的图形环境中要 JOIN 哪些字段和要 UNION 哪些字段。一般而言，在两个数据源之间的数据的实例之间有对应关系的情况下，JOIN 将比 UNION 优先。另一方面，当对于两个数据源定义相同的数据项，但在实例值中几乎没有重叠时，UNION 优先。在前述的示例中，用户可能已经 UNION 了选择的字段，认为每一个 UNION 的字段相类似，因此，查询的结果应该返回每一个 UNION 的两个字段。虽然返回两个字段，字段在一个列中彼此附加到一起，因此，作为一个可查看的字段返回。将删除重复的数据。

除了给用户提供 JOIN 或 UNION 字段的灵活性之外，用户还能够对结果进行排序。即，根据表 1404 的各个单元中的字段的顺序（即，从左到右）对结果的顺序进行排序。如此，用户可以操纵每一个可用的结果字段的相对位置，以返回所希望的结果，并控制返回结果的顺序。

应该注意，上面的实施例只是说明性的，而不作出限制。那些精通本技术的人员将会想出本发明的范围内的其他实施例。例如，在一个实施例中，表 1404 可以用选定的字段“预先填充”。如此，管理员可以指定用户选择的字段的特定排列。或者，用户可以保存字段的特定排列。作为示例，在点击 GUI 1300（图 12）的“下一步”按钮 1306 之后，可以向用户呈现图 14 的 GUI 1400，其中，已经为“用户”、“地址”和“临床信息”分配了第一行的默认单元。虽然未显示，但是，也可以预先填充多行，默认情况下，可以 UNION 一些单元。然后，用户可以排列列表 1402 中的其余可用的结果字段的相对位置（即，“位置”和“病人”），甚至可以改变默认布局（即，“用户”、“地址”和“临床信息”）。在这方面，需要注意的是，管理员可以不必位于最佳的位置来判断哪些字段可以被 UNION，假定他们可能不会对基础数据有充分的理解。相应地，可以就某些字段是否应该被 UNION（即，在查询创建表 1404 中垂直地相关）还是水平地 JOIN 作出最佳猜测估计。例如，可以通过一个算法来运行 DAM，以判断哪些逻辑字段彼此类似，无论是由于同义还是具有相同的数据类型和长度，可以完成最佳猜测估计。例如，包含病人信息的两个表可能具有非常类似的架构，子表/超表定义（一个表比另一个表具有更多信息）除外。相比之下，病人表和雇员表具有一些通用信息（例如，姓名、地址），但每一个表都相对于另一个表具有大量的不同信息（例如，职务说明、经理、病人 ID）。最佳猜测估计算法将认识到，病人和雇员表应该被 JOIN，但两个病人表应该被 UNION。

一旦排列结果字段以返回所希望的结果，用户点击 OK 按钮 1502，如图 15 所示。在点击 OK 按钮 1502 时，从查询创建表 1404 中删除任何空列和/或行。然后，将用户返回到 GUI 800 的刷新的实例，如图 16 所示。注意，摘要部分 920 现在包括如图 15 的表 1404 中定义的结果字段规范 1604。如果用户对构建的查询表示满意，则用户可以通过按下“运行搜索”按钮 924 来执行查询。具体来说，按下“运行搜索”按钮 924 会调用运行时组件 150，哪些运行时组件开始执行。

在一个实施例中，按下“运行搜索”按钮 924 首先使摘要部分 920 中的指定的查询以抽象形式呈现。然后，以上文参考图 3 和 4 所描述的方式执行所产生的抽象查询。

下面将参考图 17 描述基于用户指定的结果字段布局（例如，如图 15 的表 1404 所指定的）呈现抽象查询的方法 1700 的一个实施例。为了便于说明，假设查询创建表 1404 不包括任何空列或行，虽然单个单元可以是空的。方法 1700 开始处理查询创建表 1404 中的第一行。具体来说，方法 1700 进入循环（在步骤 1702 中），该循环对于查询创建表 1404 中的第一行中的每一个单元重复执行。检查单元，以判断它是否包含数据，即，结果字段（步骤 1704）。如果不，方法 1700 返回到步骤 1702，以开始处理第一行的下一单元。然而，对于包含结果字段的单元，作为抽象查询的组件添加结果字段名称（步骤 1706），然后，处理过程返回到步骤 1702。一旦根据在步骤 1702 中进入的循环处理完第一行中的每一个单元，方法 1700 进入步骤 1708。

在步骤 1708 中，方法 1700 对于查询创建表 1404 中的每一个其余行的每一个单元进入循环（循环在步骤 1708 和步骤 1710 中进入的循环）。对于给定行的给定单元，方法 1700 判断单元是否包含结果字段（步骤 1712）。如果不，方法返回到步骤 1710 以开始处理下一单元。然而，对于确实包含结果字段的给定单元，方法 1700 判断给定单元紧上面的相邻单元是否包含数据（步骤 1714）。如果是，则将两个单元 UNION 起来（步骤 1716）。一旦处理完每一个其余的行中的每一个单元，方法 1700 退出。如此，每一个垂直相关的单元（即，列中的相邻单元）被 UNION。

方法 1700 的结果是抽象查询。只作为说明，下面显示了对应于图 15 的查询创建表 1404 的典型的抽象查询。

具有 UNION 的抽象查询的示例

```
<QueryAbstraction version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="QueryAbstraction.xsd">
    <Selection>
        <Condition relOperator="AND" field="data://patient/User">
            <Value val="John"/>
        </Condition>
    </Selection>
    <Results format="HTML" distinct="Yes" >
        <Union>
            <Field name="data://patient/User" fieldType="char"/>
            <Field name="data://patient/Patient" fieldType="char"/>
        </Union>
        <Union>
            <Field name="data://patient/Address" fieldType="char"/>

            <Field name="data://patient/Location" fieldType="char"/>
        </Union>
        <Field name="data://patient/Clinial Info" fieldType="char"/>
    </Results>
</QueryAbstraction>
```

然后，可以以基本上类似于参考图 3-4 所描述的方式将上述抽象查询转换成具体查询。运行时组件 150 将按预期的那样实现将数据返回到最终用户所需要的适当的 UNION 和 JOIN。上面提供了抽象查询的结果字段部分的 SQL 表示（即，对应于 SQL 语句的 SELECT 语句的那一部分）。

尽管前述的内容是针对本发明的实施例进行说明的，但是，在不偏离本发明的基本范围的情况下，可以研究出本发明的其他实施例，其范围由随后的权利要求来确定。

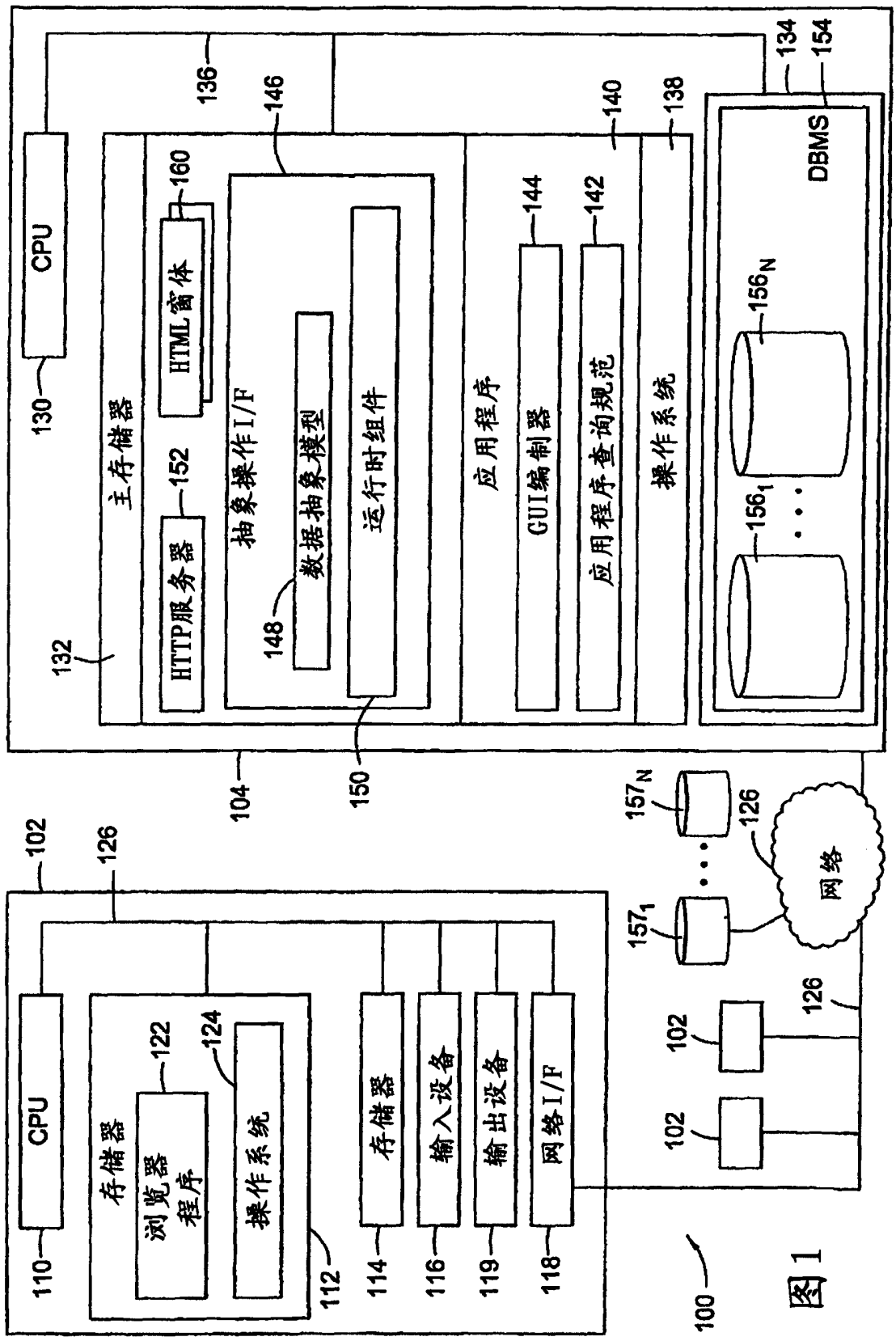

CPU
130
主存储器
132
HTTP服务器
152
HTML窗体
160
抽象操作I/F
146
数据抽象模型
148
运行时组件
150
应用程序
GUI编制器
144
应用程序查询规范
142
操作系统
140
DBMS
138
134
154
136
104
102
126
CPU
110
存储器
浏览器程序
122
操作系统
124
112
存储器
114
输入设备
116
输出设备
119
网络I/F
118
157_N
157_1
156_N
156_1
网络
100

图 1

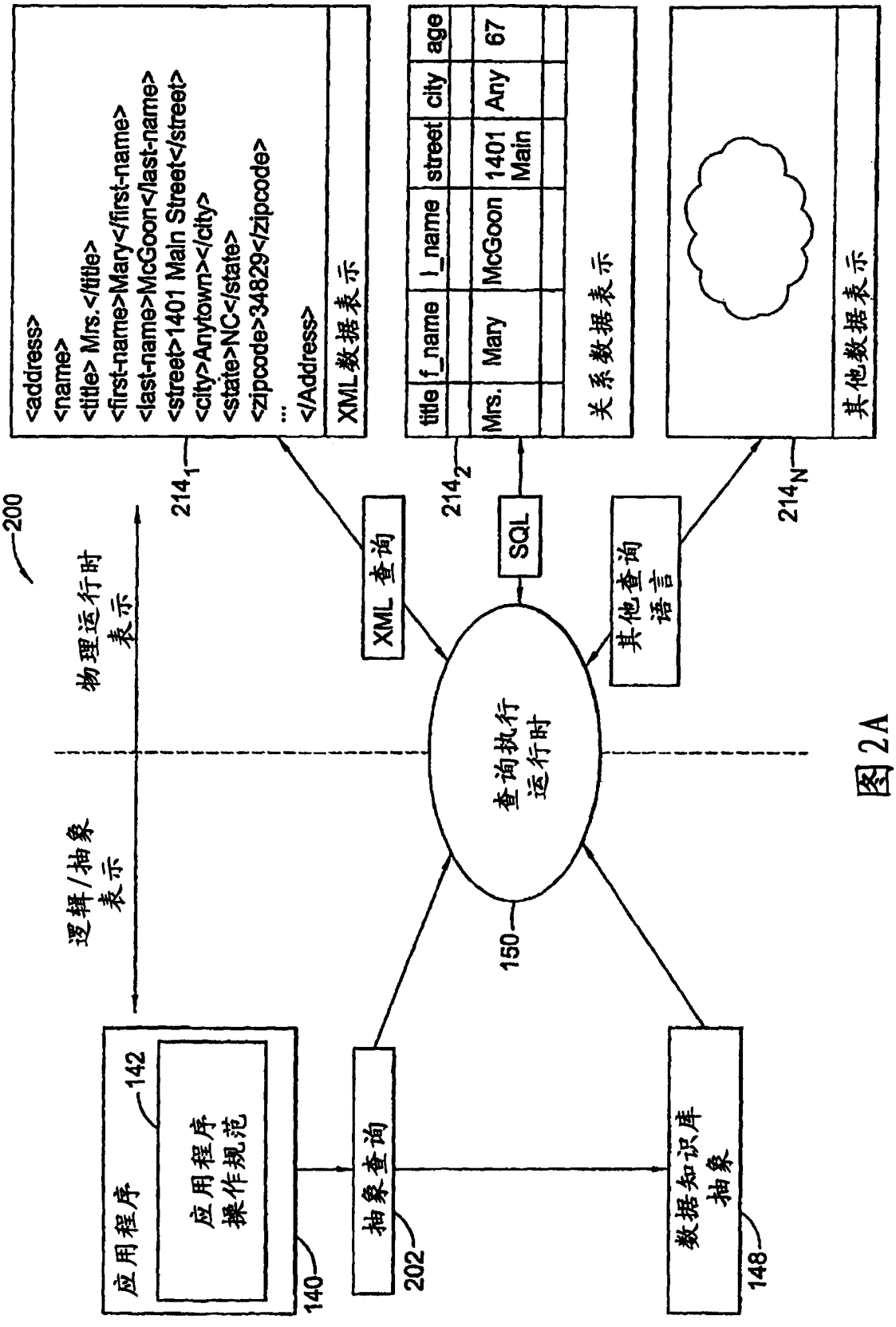

200
物理运行时表示
逻辑/抽象表示
<address>
<name>
<title> Mrs.</title>
<first-name>Mary</first-name>
<last-name>McGoon</last-name>
<street>1401 Main Street</street>
<city>Anytown></city>
<state>NC</state>
<zipcode>34829</zipcode>
...
</Address>
XML数据表示
2141
title
f_name
l_name
street
city
age
Mrs.
Mary
McGoon
1401 Main
Any
67
关系数据表示
2142
其他数据表示
214N
XML 查询
SQL
其他查询语言
查询执行运行时
150
应用程序
142
应用程序操作规范
140
抽象查询
202
数据知识库抽象
148

图 2A

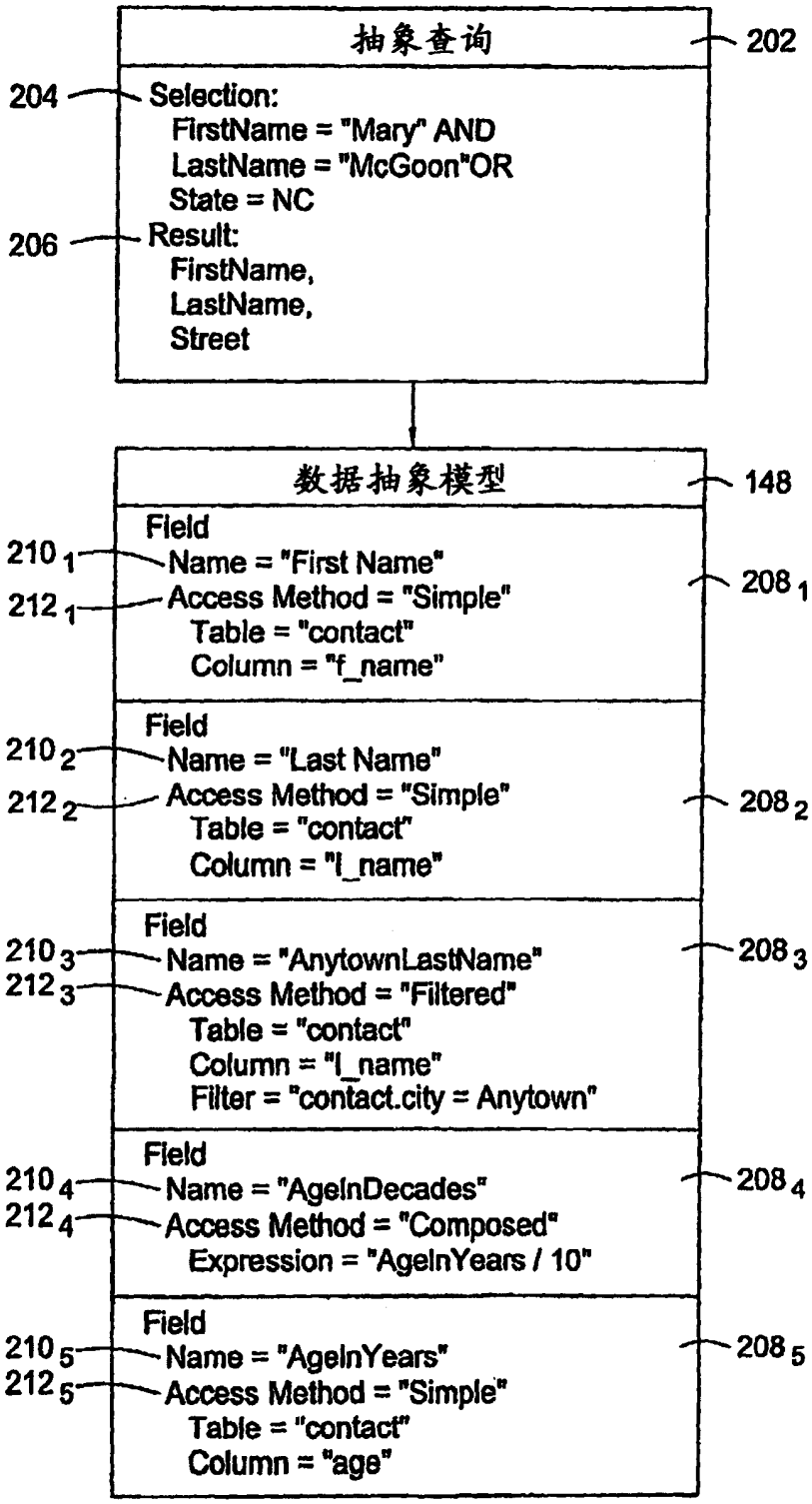

抽象查询
202
204
Selection:
FirstName = "Mary" AND
LastName = "McGoon"OR
State = NC
206
Result:
FirstName,
LastName,
Street
数据抽象模型
148
Field
210 1
Name = "First Name"
208 1
212 1
Access Method = "Simple"
Table = "contact"
Column = "f_name"
Field
210 2
Name = "Last Name"
212 2
Access Method = "Simple"
208 2
Table = "contact"
Column = "l_name"
Field
210 3
Name = "AnytownLastName"
208 3
212 3
Access Method = "Filtered"
Table = "contact"
Column = "l_name"
Filter = "contact.city = Anytown"
Field
210 4
Name = "AgeInDecades"
208 4
212 4
Access Method = "Composed"
Expression = "AgeInYears / 10"
Field
210 5
Name = "AgeInYears"
208 5
212 5
Access Method = "Simple"
Table = "contact"
Column = "age"

图 2B

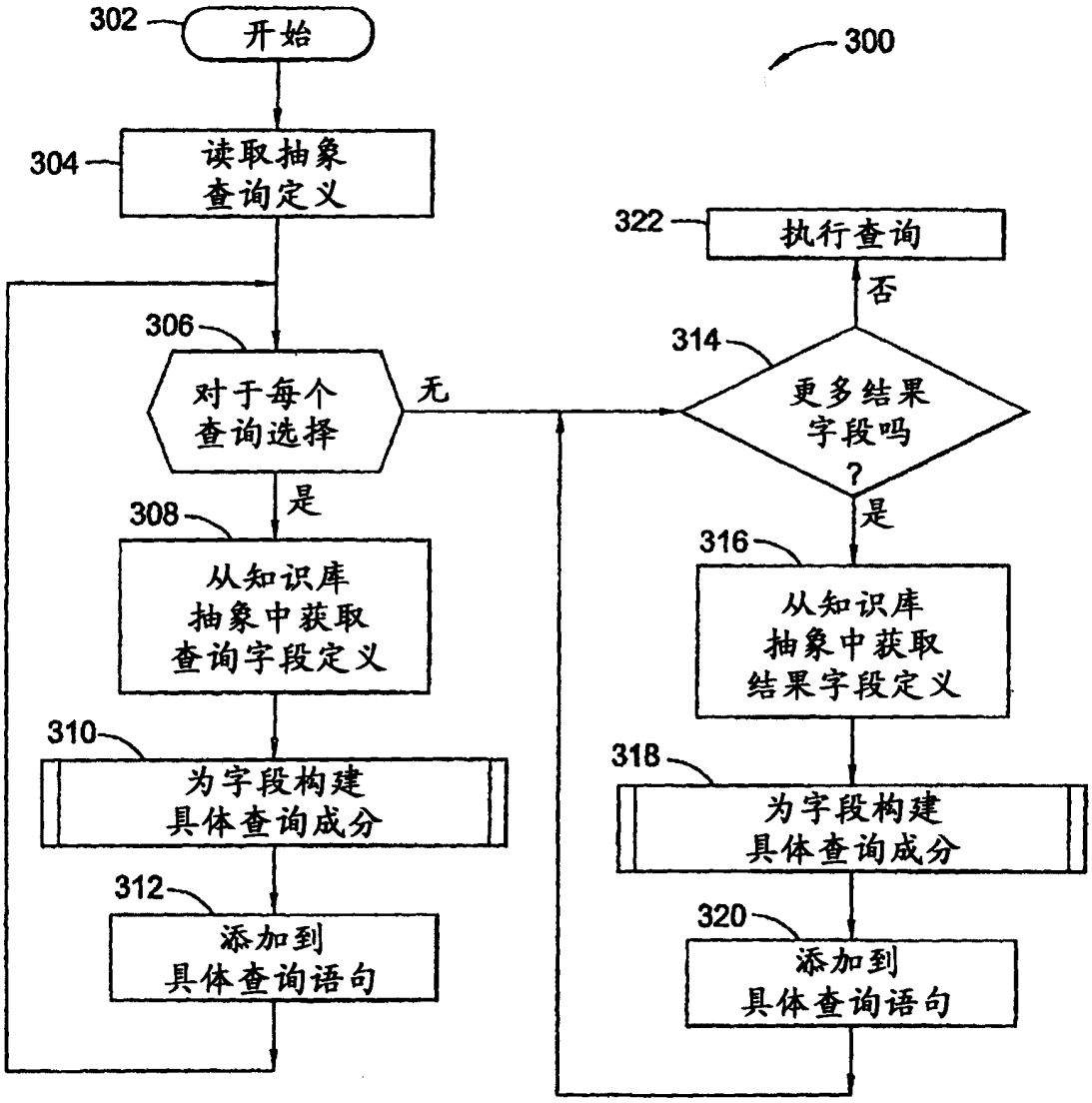

302
开始
300
304
读取抽象
查询定义
322
执行查询
否
306
对于每个
查询选择
无
314
更多结果
字段吗
?
是
308
从知识库
抽象中获取
查询字段定义
316
从知识库
抽象中获取
结果字段定义
310
为字段构建
具体查询成分
318
为字段构建
具体查询成分
312
添加到
具体查询语句
320
添加到
具体查询语句

图3

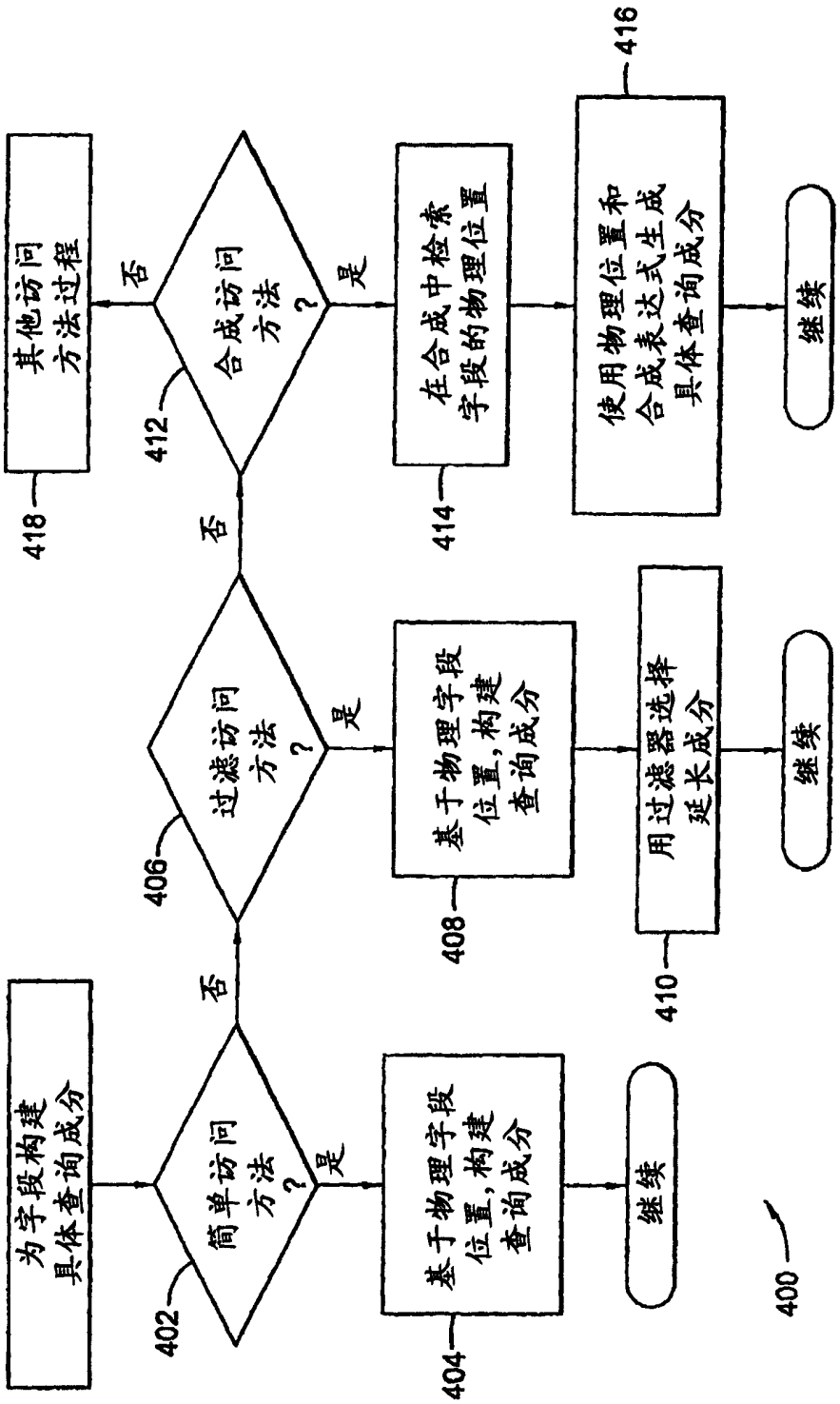

为字段构建具体查询成分
简单访问方法？
是
否
基于物理字段位置，构建查询成分
继续
过滤访问方法？
是
否
基于物理字段位置，构建查询成分
用过滤器选择延长成分
继续
合成访问方法？
是
否
在合成中检索字段的物理位置
使用物理位置和合成表达式生成具体查询成分
继续
其他访问方法过程
400
402
404
406
408
410
412
414
416
418

图 4

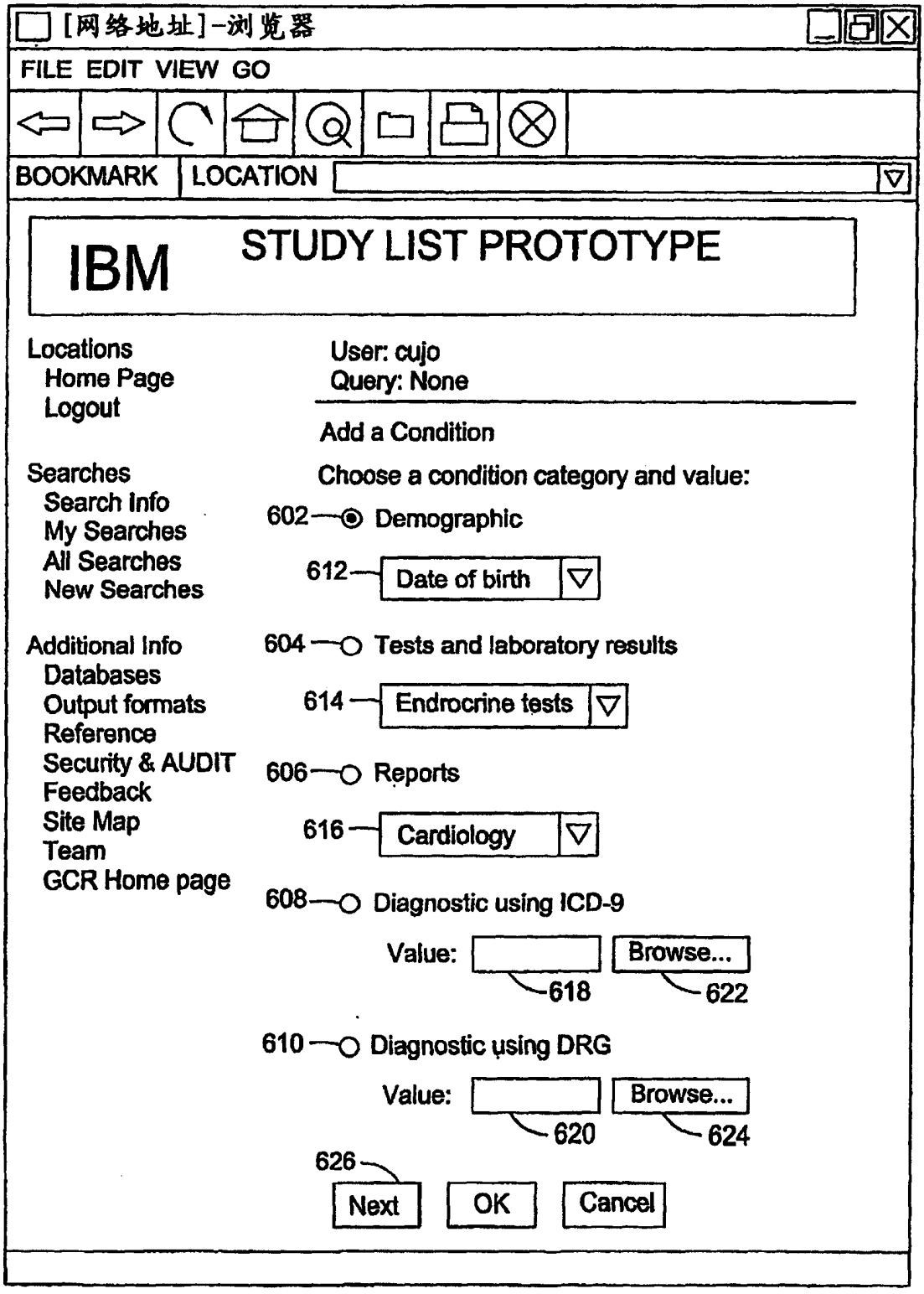

[网络地址]-浏览器
FILE EDIT VIEW GO
BOOKMARK
LOCATION
IBM
STUDY LIST PROTOTYPE
Locations
Home Page
Logout
Searches
Search Info
My Searches
All Searches
New Searches
Additional Info
Databases
Output formats
Reference
Security & AUDIT
Feedback
Site Map
Team
GCR Home page
User: cujo
Query: None
Add a Condition
Choose a condition category and value:
602
Demographic
612
Date of birth
604
Tests and laboratory results
614
Endrocrine tests
606
Reports
616
Cardiology
608
Diagnostic using ICD-9
Value:
Browse...
618
622
610
Diagnostic using DRG
Value:
Browse...
620
624
626
Next
OK
Cancel
600

图 5

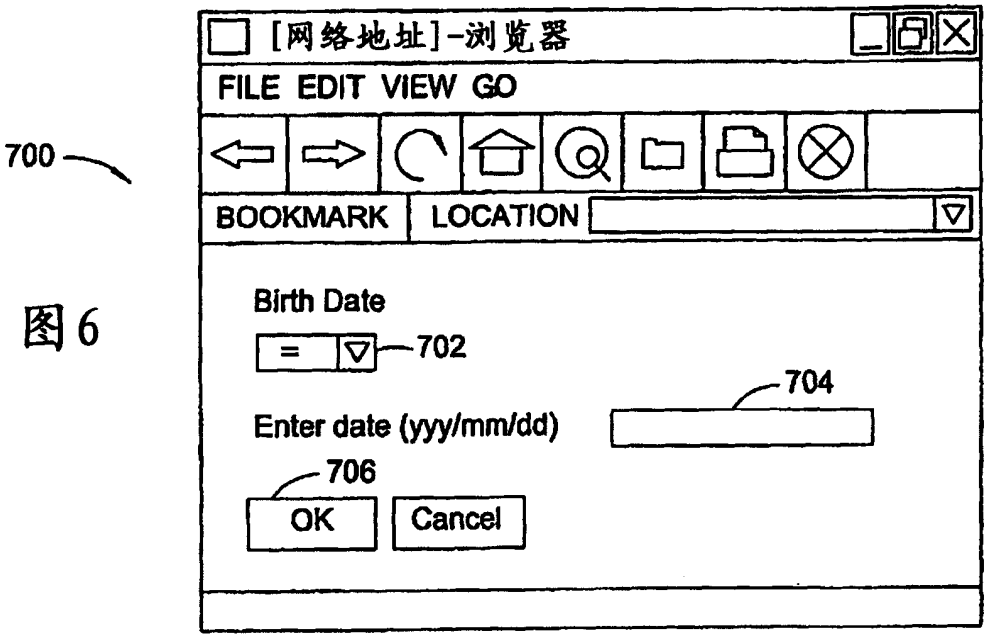

[网络地址]-浏览器
FILE EDIT VIEW GO
BOOKMARK
LOCATION
Birth Date
=
702
704
Enter date (yyy/mm/dd)
706
OK
Cancel
700

图 6

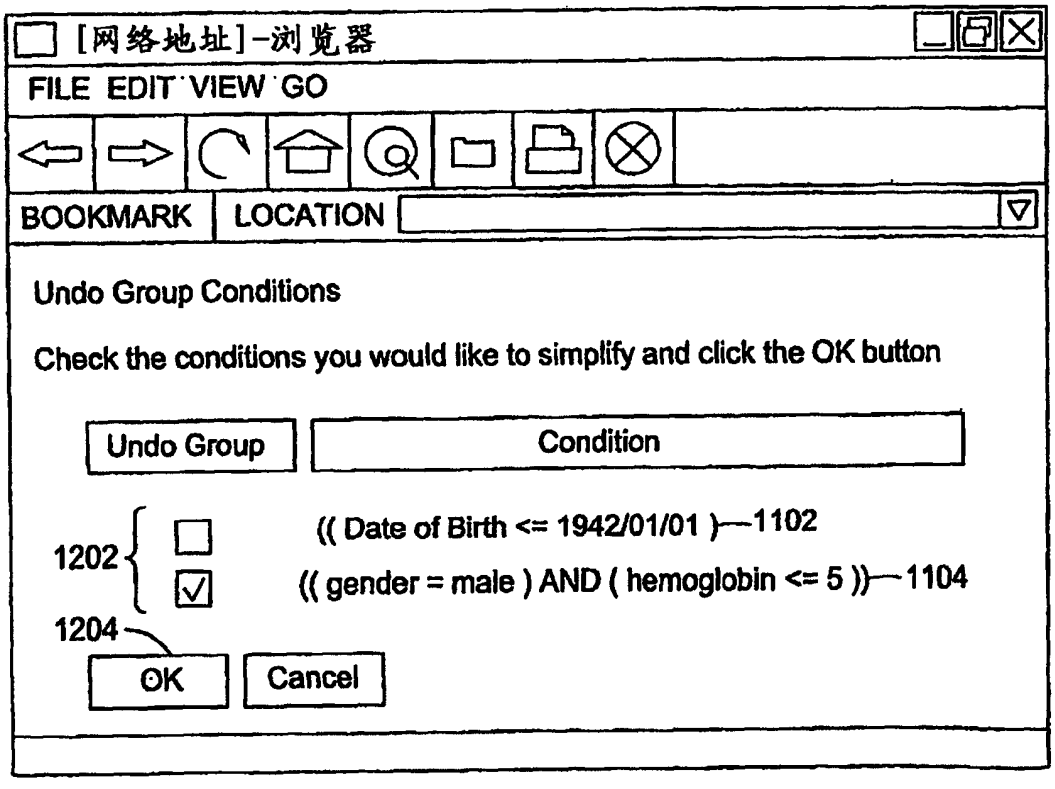

[网络地址]-浏览器
FILE EDIT VIEW GO
BOOKMARK
LOCATION
Undo Group Conditions
Check the conditions you would like to simplify and click the OK button
Undo Group
Condition
1202
((Date of Birth <= 1942/01/01)
1102
((gender = male) AND (hemoglobin <= 5))
1104
1204
OK
Cancel
1200

图 11

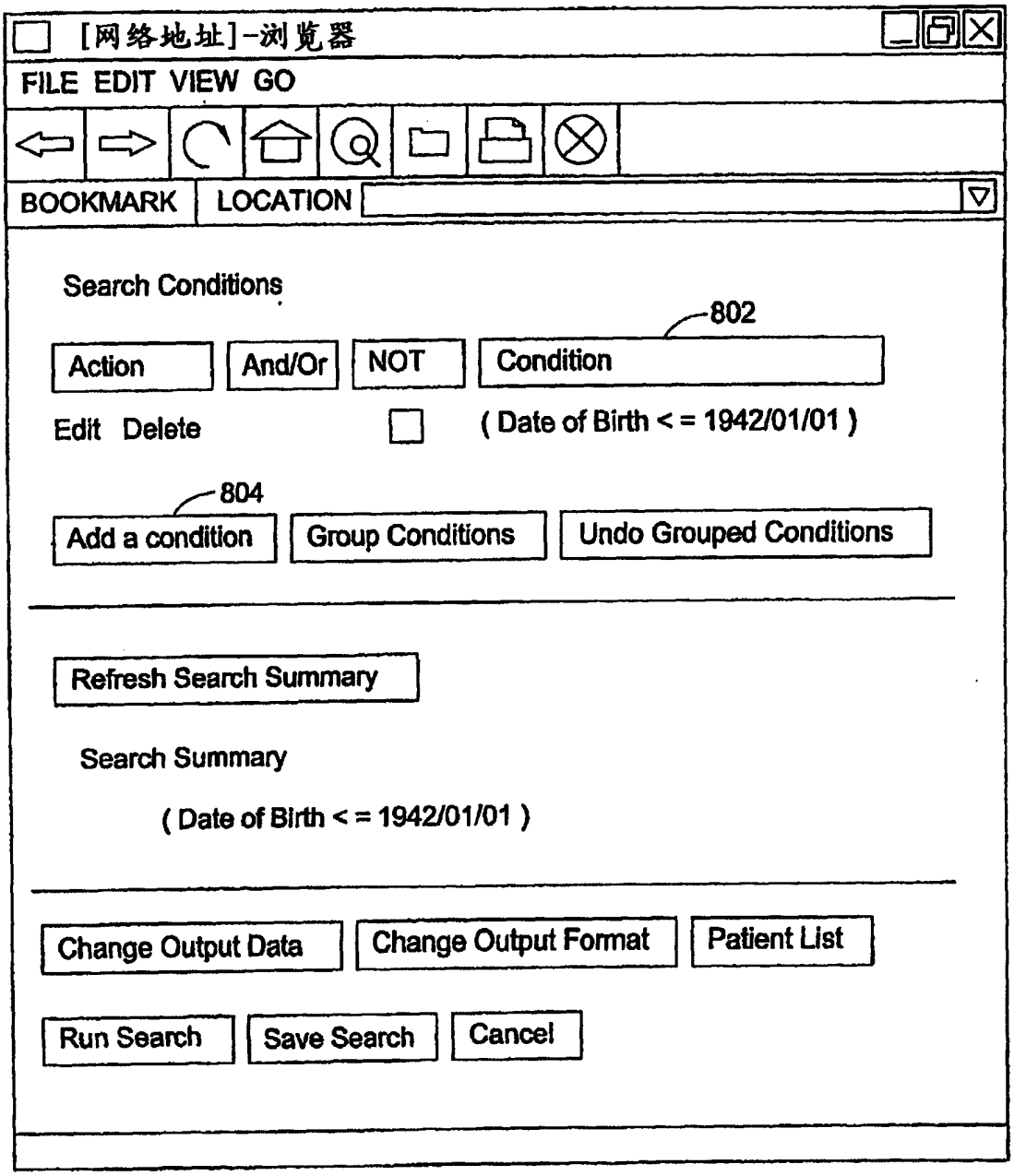

[网络地址]-浏览器
FILE EDIT VIEW GO
BOOKMARK
LOCATION
Search Conditions
802
Action
And/Or
NOT
Condition
Edit Delete
(Date of Birth < = 1942/01/01)
804
Add a condition
Group Conditions
Undo Grouped Conditions
Refresh Search Summary
Search Summary
(Date of Birth < = 1942/01/01)
Change Output Data
Change Output Format
Patient List
Run Search
Save Search
Cancel
800

图7

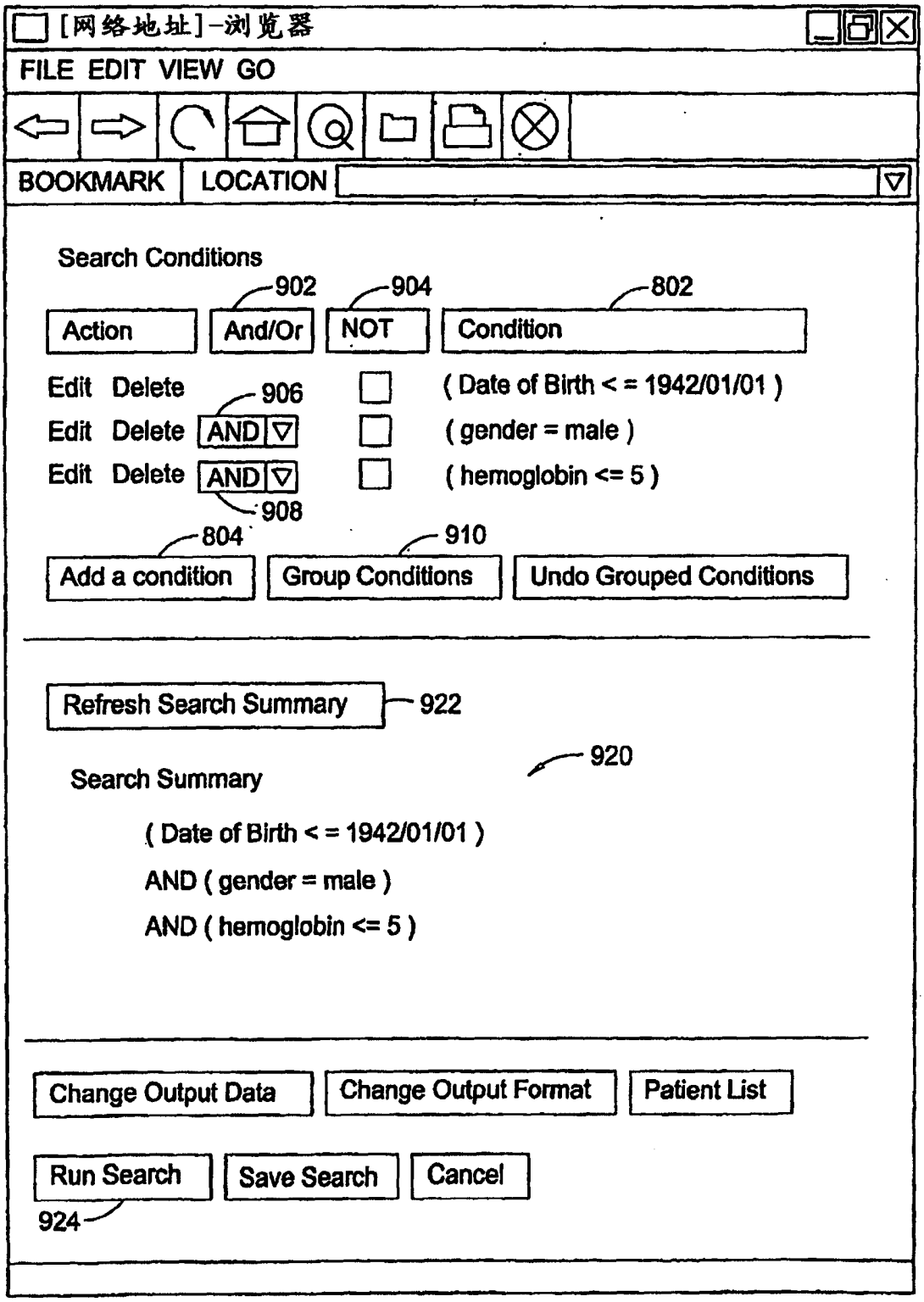

[网络地址]-浏览器
FILE EDIT VIEW GO
BOOKMARK
LOCATION
Search Conditions
902
904
802
Action
And/Or
NOT
Condition
Edit Delete
906
(Date of Birth < = 1942/01/01)
Edit Delete
AND
(gender = male)
Edit Delete
AND
(hemoglobin <= 5)
908
804
910
Add a condition
Group Conditions
Undo Grouped Conditions
Refresh Search Summary
922
920
Search Summary
(Date of Birth < = 1942/01/01)
AND (gender = male)
AND (hemoglobin <= 5)
Change Output Data
Change Output Format
Patient List
Run Search
Save Search
Cancel
924
800

图 8

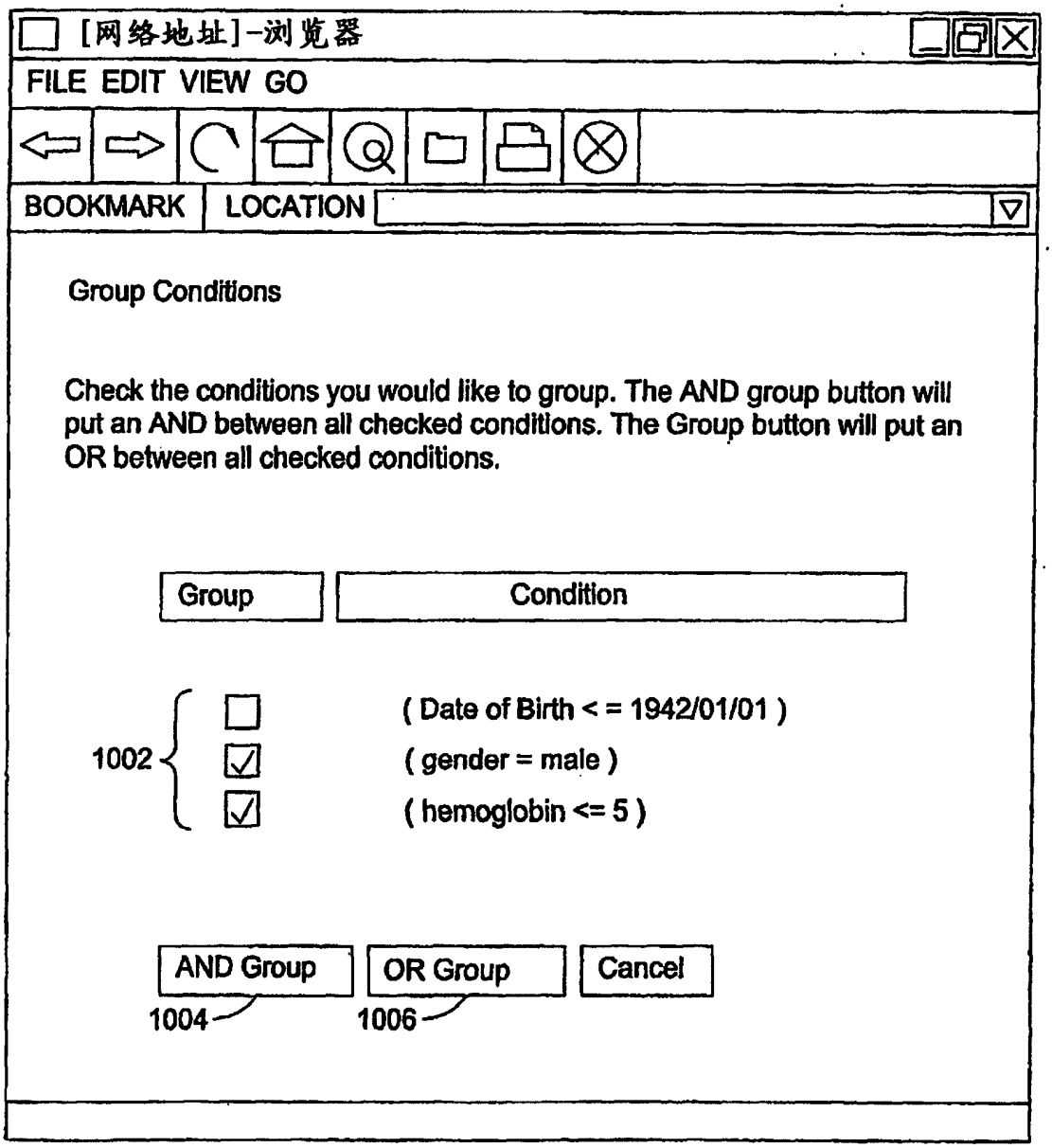

[网络地址]-浏览器
FILE EDIT VIEW GO
BOOKMARK
LOCATION
Group Conditions
Check the conditions you would like to group. The AND group button will put an AND between all checked conditions. The Group button will put an OR between all checked conditions.
Group
Condition
1002
(Date of Birth < = 1942/01/01)
(gender = male)
(hemoglobin <= 5)
AND Group
OR Group
Cancel
1004
1006
1000

图 9

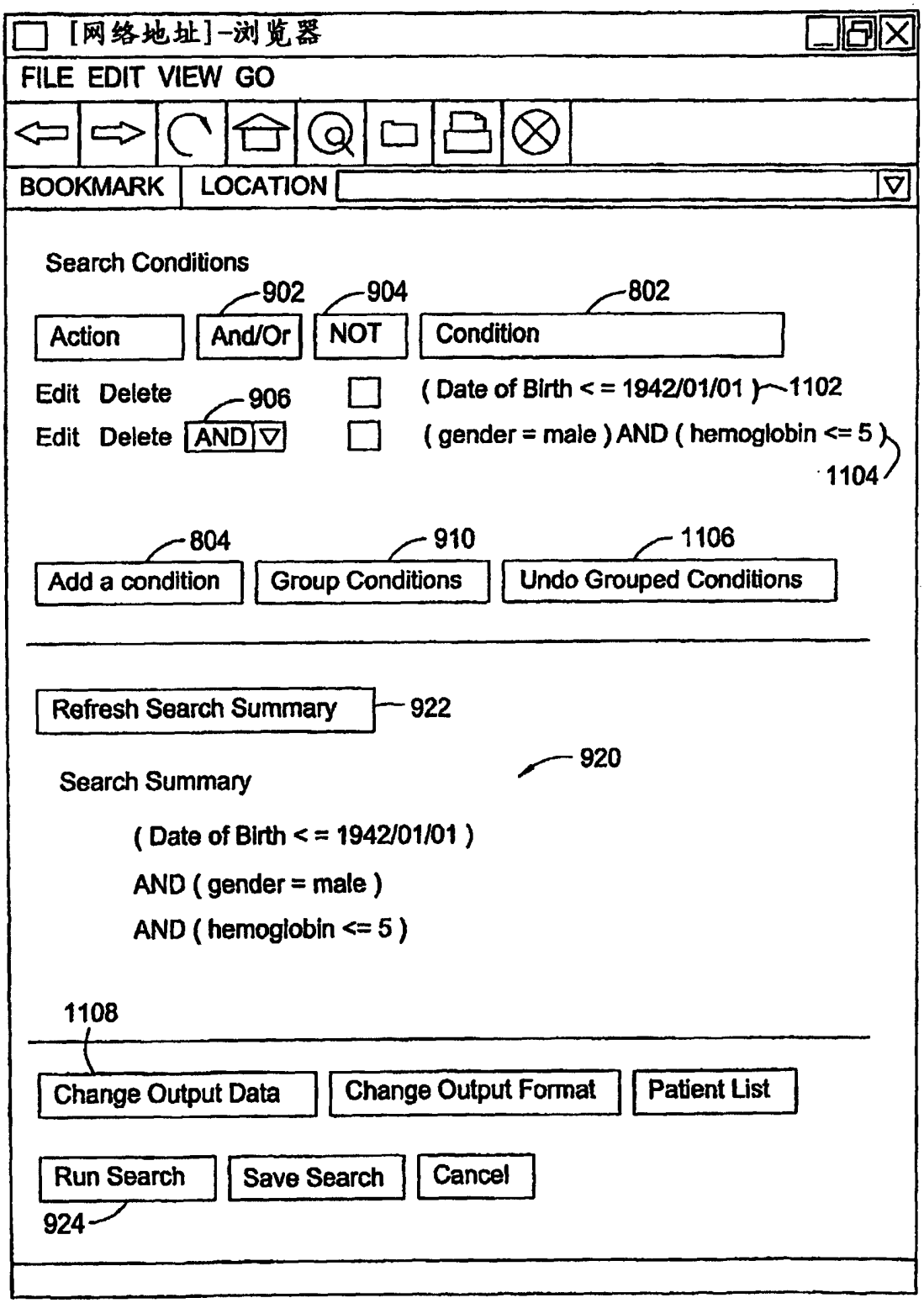

[网络地址]-浏览器
FILE EDIT VIEW GO
BOOKMARK
LOCATION
Search Conditions
902
904
802
Action
And/Or
NOT
Condition
Edit Delete
906
(Date of Birth < = 1942/01/01)
1102
Edit Delete
AND
(gender = male) AND (hemoglobin <= 5)
1104
804
910
1106
Add a condition
Group Conditions
Undo Grouped Conditions
Refresh Search Summary
922
920
Search Summary
(Date of Birth < = 1942/01/01)
AND (gender = male)
AND (hemoglobin <= 5)
1108
Change Output Data
Change Output Format
Patient List
Run Search
Save Search
Cancel
924

图10

800

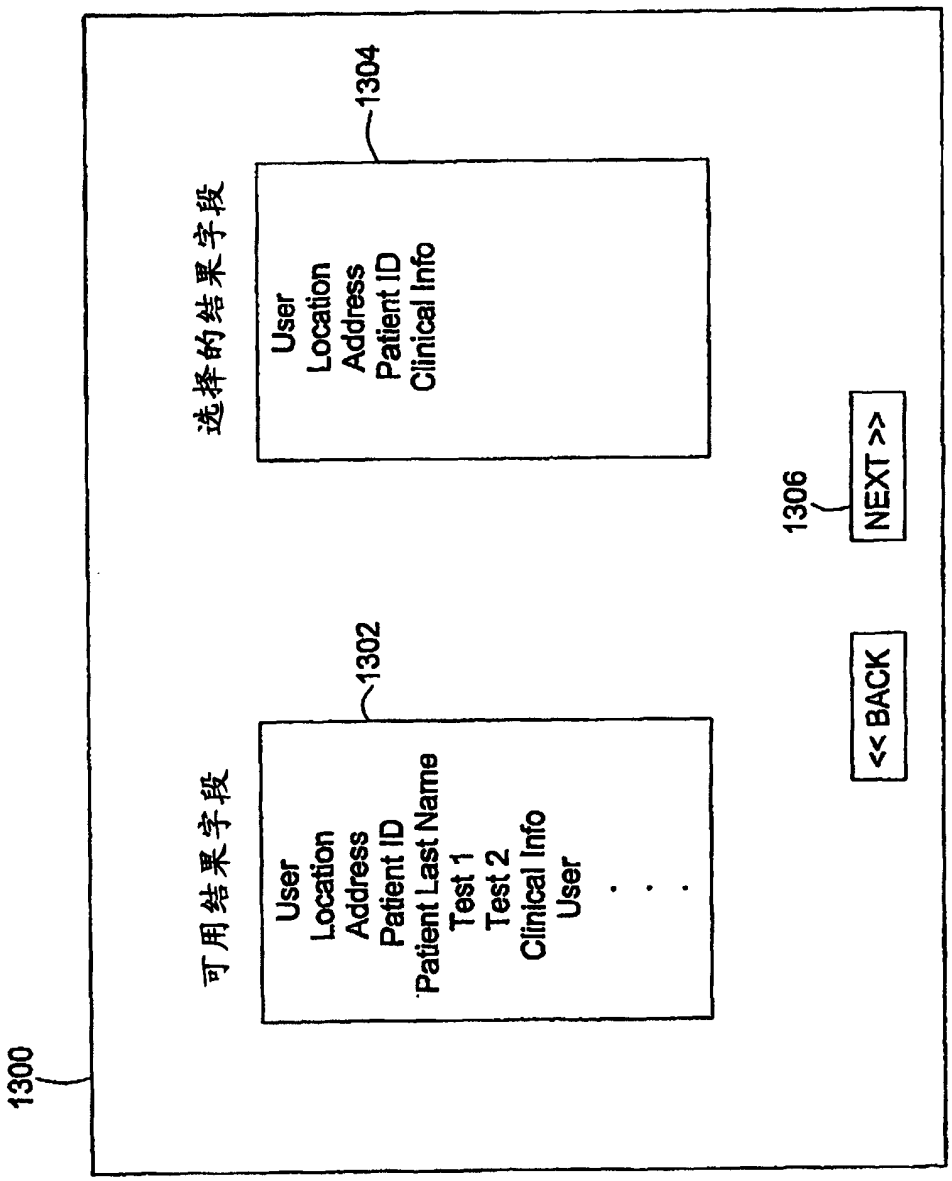

1300
可用结果字段
1302
User
Location
Address
Patient ID
Patient Last Name
Test 1
Test 2
Clinical Info
User
选择的结果字段
1304
User
Location
Address
Patient ID
Clinical Info
<< BACK
1306
NEXT >>

图 12

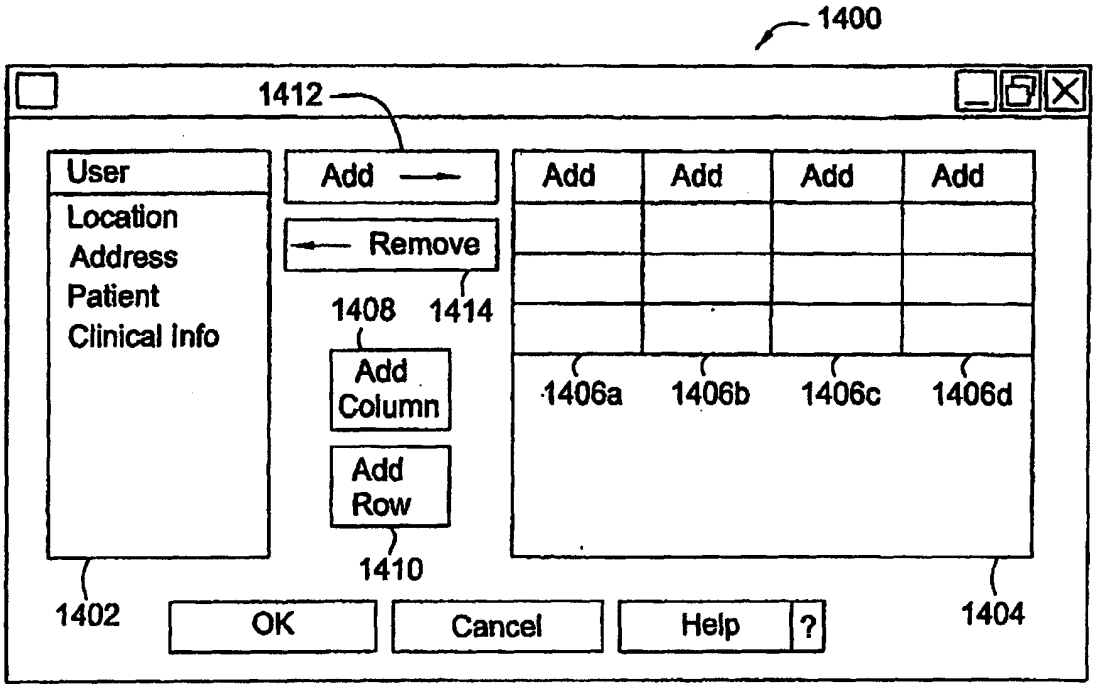

1400
1412
User
Location
Address
Patient
Clinical Info
Add
Remove
1408
1414
Add Column
Add Row
1410
1402
Add
Add
Add
Add
1406a
1406b
1406c
1406d
1404
OK
Cancel
Help
?

图13

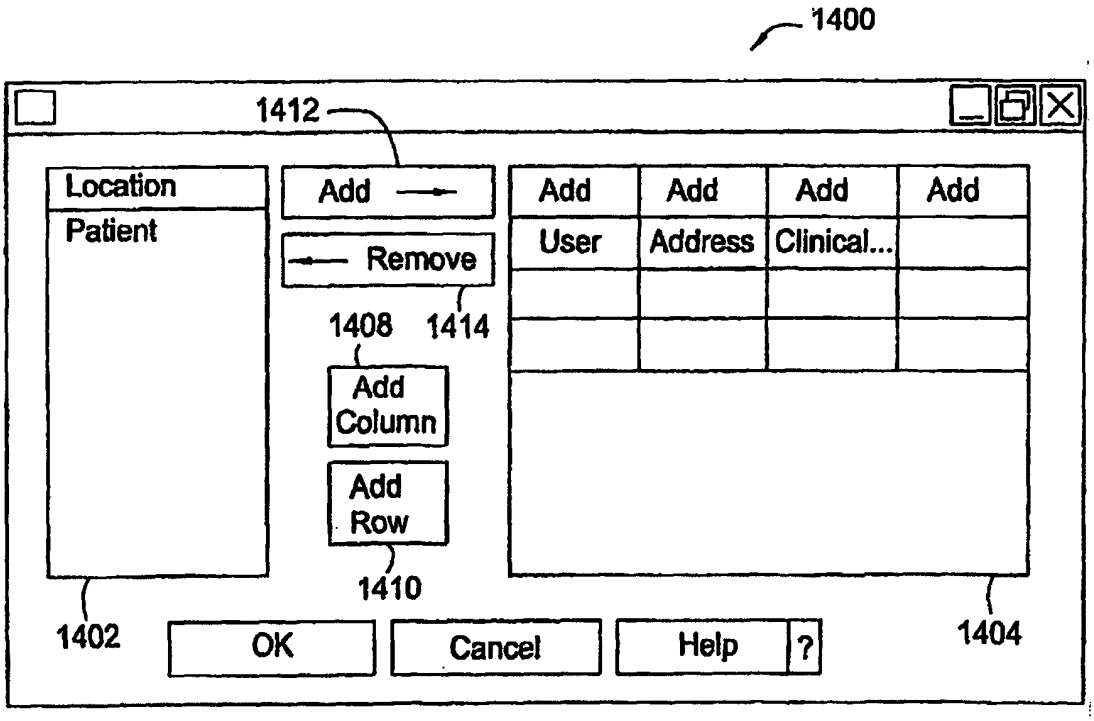

1400
1412
Location
Patient
Add
Remove
1408
1414
Add Column
Add Row
1410
1402
Add
Add
Add
Add
User
Address
Clinical...
1404
OK
Cancel
Help
?

图14

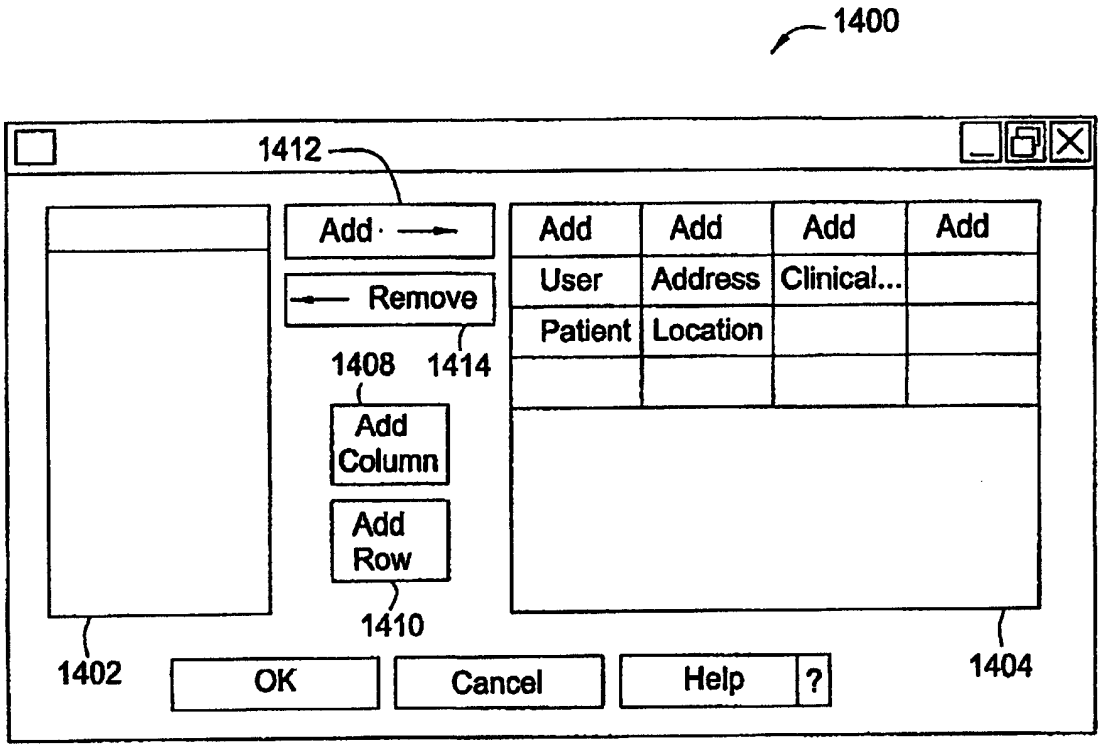

1400
1412
Add
Remove
1408
1414
Add Column
Add Row
1410
1402
OK
Cancel
Help
?
Add
Add
Add
Add
User
Address
Clinical...
Patient
Location
1404

图15

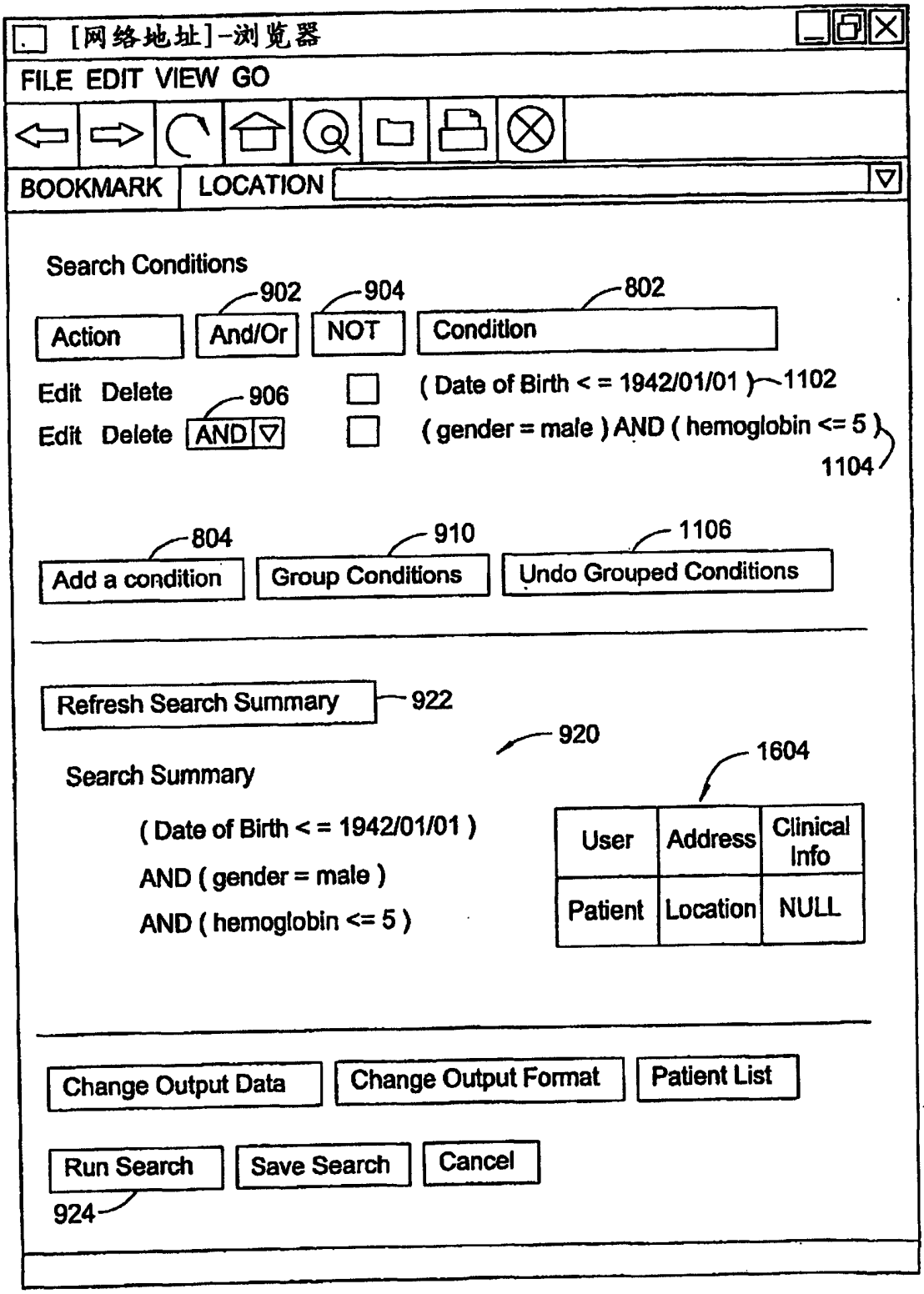
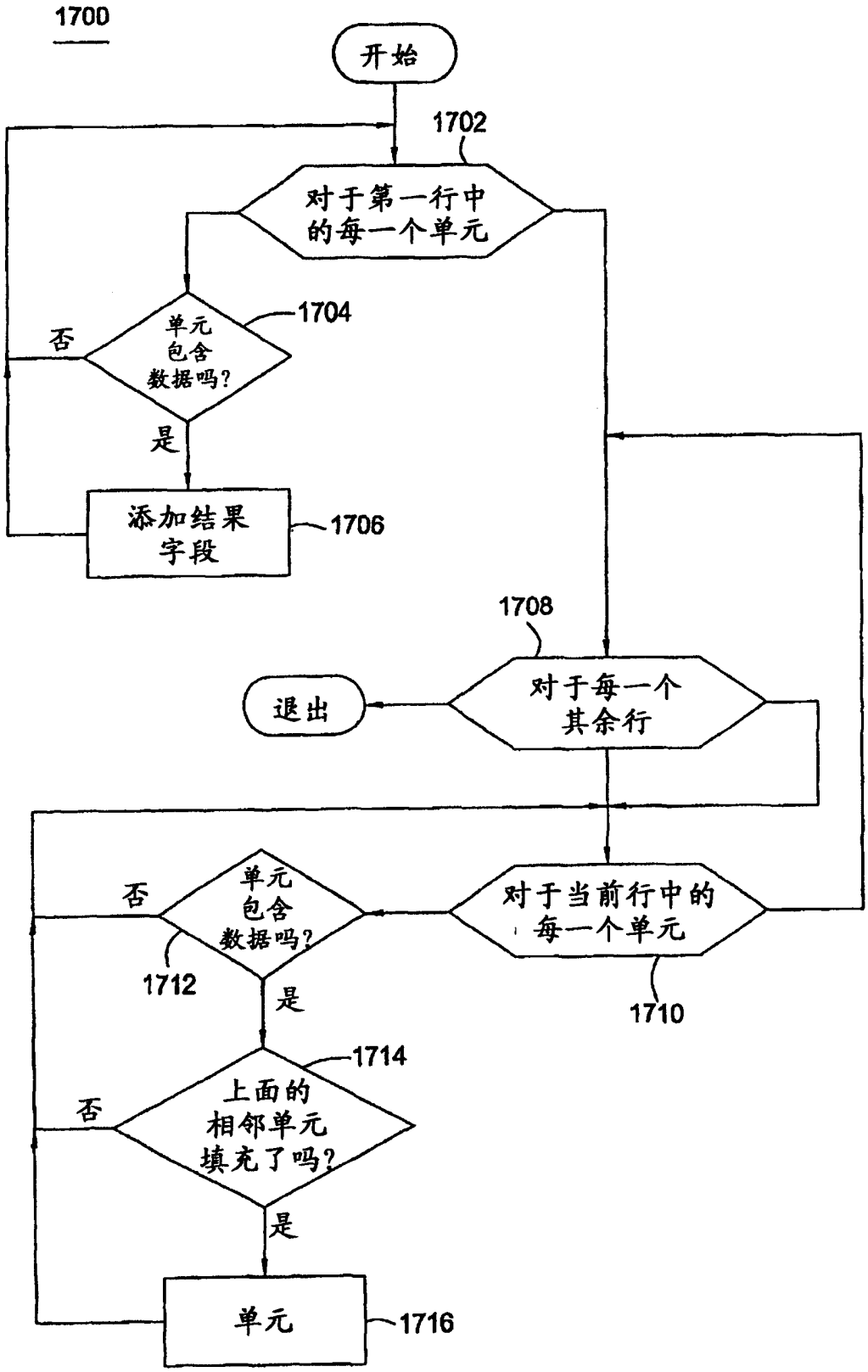

[网络地址]-浏览器
FILE EDIT VIEW GO
BOOKMARK
LOCATION
Search Conditions
902
904
802
Action
And/Or
NOT
Condition
Edit Delete
906
(Date of Birth < = 1942/01/01)
1102
Edit Delete
AND
(gender = male) AND (hemoglobin <= 5)
1104
804
910
1106
Add a condition
Group Conditions
Undo Grouped Conditions
Refresh Search Summary
922
920
1604
Search Summary
(Date of Birth < = 1942/01/01)
AND (gender = male)
AND (hemoglobin <= 5)
User
Address
Clinical Info
Patient
Location
NULL
Change Output Data
Change Output Format
Patient List
Run Search
Save Search
Cancel
924
800

图16

图17


1700
开始
1702
对于第一行中的每一个单元
1704
单元包含数据吗?
否
是
添加结果字段
1706
1708
退出
对于每一个其余行
单元包含数据吗?
否
1712
是
对于当前行中的每一个单元
1710
1714
上面的相邻单元填充了吗?
否
是
单元
1716