



(12) 发明专利

(10) 授权公告号 CN 107257973 B

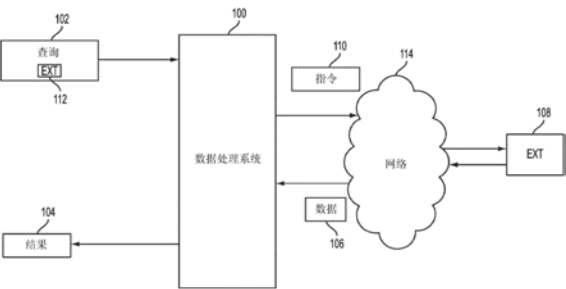
(45) 授权公告日 2022.03.04

(21) 申请号 201680011145.9  
(22) 申请日 2016.02.16  
(65) 同一申请的已公布的文献号  
    申请公布号 CN 107257973 A  
(43) 申请公布日 2017.10.17  
(30) 优先权数据  
    62/117,588 2015.02.18 US  
    14/752,094 2015.06.26 US  
(85) PCT国际申请进入国家阶段日  
    2017.08.18  
(86) PCT国际申请的申请数据  
    PCT/US2016/018028 2016.02.16  
(87) PCT国际申请的公布数据  
    W02016/133880 EN 2016.08.25  
(73) 专利权人 起元技术有限责任公司  
    地址 美国马萨诸塞州

(72) 发明人 I.谢克特 G.J.艾琳  
(74) 专利代理机构 北京林达刘知识产权代理事务  
    所(普通合伙) 11277  
    代理人 刘新宇  
(51) Int.Cl.  
    G06F 16/25 (2019.01)  
    G06F 16/2452 (2019.01)  
(56) 对比文件  
    US 2009055370 A1,2009.02.26  
    US 2009055370 A1,2009.02.26  
    US 2001011371 A1,2001.08.02  
    US 2006294075 A1,2006.12.28  
    US 2013246864 A1,2013.09.19  
    CN 1541363 A,2004.10.27  
    CN 101535990 A,2009.09.16  
审查员 张书旋  
权利要求书8页 说明书17页 附图14页

(54) 发明名称  
    查询网络上的数据源

(57) 摘要  
    描述了一种利用数据处理系统所进行的计算机实现方法,用于基于根据能够适用于关系数据库的查询语言所表示的查询来执行计算机程序,所述计算机程序是至少部分地基于存储在有形、非暂时性计算机可读介质中的数据来执行的,其中,执行所述计算机程序的步骤包括:接收SQL查询,其中所述SQL查询包括与所述数据处理系统外部的资源相关联的标识符;基于所述SQL查询来生成计算机程序;以及执行所述计算机程序,使得:向所述资源发送一个或多个指令,所述指令定义了除所述SQL查询的操作以外的操作;以及响应于所述指令而从所述资源接收数据。



1. 一种利用数据处理系统所进行的计算机实现方法,用于基于根据能够适用于关系数据库的查询语言所表示的查询来执行计算机程序,所述计算机程序是至少部分地基于存储在有形、非暂时性计算机可读介质中的数据来执行的,其中,执行所述计算机程序的步骤包括:

接收SQL查询,其中所述SQL查询包括与所述数据处理系统外部的资源相关联的标识符;

基于所述SQL查询来生成计算机程序,包括:

识别包括与所述资源相关联的至少一个条目的目录,所述至少一个条目包括表示能够用于对计算机程序进行配置以访问所述资源的数据的配置数据的一个或多个数据值,以及从所识别的目录接收所述配置数据;

基于所述配置数据来执行所述计算机程序,所述计算机程序的执行使得:

向所述资源发送一个或多个指令,所述指令定义了除所述SQL查询的操作以外的操作,以及

响应于所述指令而从所述资源接收数据。

2. 根据权利要求1所述的计算机实现方法,其中,所述计算机程序包括定义了访问所述资源的方式的可执行指令,所述可执行指令基于被提供至所述计算机程序的配置数据而操作。

3. 根据权利要求1所述的计算机实现方法,其中,在能够基于所述资源所使用的数据格式的变化而更新的目录中指定所述配置数据。

4. 根据权利要求1所述的计算机实现方法,其中,还包括:

基于所述配置数据来生成参数值;以及

向所述计算机程序的一部分提供所述参数值,该一部分能够与所述资源进行通信,所述计算机程序的执行是基于所述参数值的。

5. 根据权利要求4所述的计算机实现方法,其中,基于所述配置数据来生成参数值的步骤包括:执行用于生成所述计算机程序的所述一部分能够读取的格式的参数文件的参数生成器,以及

向所述计算机程序的所述一部分提供所述参数值的步骤包括:使所述参数文件能够用于所述计算机程序的所述一部分。

6. 根据权利要求1所述的计算机实现方法,其中,所述SQL查询包括具有自变量的SELECT语句,所述自变量的至少一部分与和所述资源相关联的所述标识符相对应。

7. 根据权利要求1所述的计算机实现方法,其中,所述计算机程序包括表示所述SQL查询的操作的组件。

8. 根据权利要求1所述的计算机实现方法,其中,所述计算机程序是数据流图,并且该数据流图的子图进行向所述资源的所述一个或多个指令的发送。

9. 根据权利要求1所述的计算机实现方法,其中,使用所述资源所公开的应用编程接口即API来访问该资源。

10. 根据权利要求9所述的计算机实现方法,其中,还包括:

使得能够响应于接收到所述SQL查询而执行所述API的功能,

所述API的功能能够在所述资源处执行,并且被发送至所述资源的指令使得能够执行

所述API的功能。

11. 根据权利要求1所述的计算机实现方法,其中,还包括:以数据库表的形式对从外部资源接收到的数据进行格式化。

12. 根据权利要求1所述的计算机实现方法,其中,所述资源不是关系数据库管理系统。

13. 根据权利要求1所述的计算机实现方法,其中,所述指令被发送至不会响应于SQL查询而返回结果的资源设施。

14. 根据权利要求1所述的计算机实现方法,其中,响应于所述指令而从所述资源接收到的数据包括所述SQL查询所指定的数据。

15. 根据权利要求1所述的计算机实现方法,其中,还包括识别响应于所述指令而从所述资源接收到的数据中的记录和字段,所述记录和字段是基于与所述数据处理系统外部的资源相关联的记录格式来识别的。

16. 根据权利要求1所述的计算机实现方法,其中,基于所述SQL查询来生成所述计算机程序的步骤包括:

基于所述SQL查询来识别数据源;

识别除管理关系数据库的系统以外的执行系统;

基于所述查询来生成对查询计划器的请求;

向所述查询计划器提供所述请求;

接收所述查询计划器基于所述请求所生成的查询计划,所述查询计划包括对管理关系数据库的系统所要进行的一个或多个步骤的描述;

生成例示包括以下内容的数据流图的数据结构:

第一节点,用于表示要执行的至少一个操作,

所述第一节点与能够被执行系统用来调用可执行程序代码以进行操作的信息相关联,

所述第一节点与能够被执行系统用来使数据能够用于所述程序代码的信息相关联,以及

所述操作是基于所述查询计划所描述的步骤而选择的,以及

至少一个链接,用于表示所述第一节点的操作的输出数据被提供作为针对所述数据流图的第二节点的操作的输入数据。

17. 根据权利要求1所述的计算机实现方法,其中,所述配置数据指定了与所述资源进行通信时所使用的至少一个值。

18. 一种数据处理系统,其能够基于根据能够适用于关系数据库的查询语言所表示的查询来执行计算机程序,所述计算机程序是至少部分地基于存储在有形、非暂时性计算机可读介质中的数据来执行的,所述数据处理系统被配置为进行包括以下内容的操作:

接收SQL查询,其中所述SQL查询包括与所述数据处理系统外部的资源相关联的标识符;

基于所述SQL查询来生成计算机程序,包括:

识别包括与所述资源相关联的至少一个条目的目录,所述至少一个条目包括表示能够用于对计算机程序进行配置以访问所述资源的数据的配置数据的一个或多个数据值,以及从所识别的目录接收所述配置数据;以及

基于所述配置数据来执行所述计算机程序,所述计算机程序的执行使得:

向所述资源发送一个或多个指令,所述指令定义了除所述SQL查询的操作以外的操作,以及

响应于所述指令而从所述资源接收数据。

19.根据权利要求18所述的数据处理系统,其中,所述计算机程序包括定义了访问所述资源的方式的可执行指令,所述可执行指令基于被提供至所述计算机程序的配置数据而操作。

20.根据权利要求18所述的数据处理系统,其中,在能够基于所述资源所使用的数据格式的变化而更新的目录中指定所述配置数据。

21.根据权利要求18所述的数据处理系统,其中,所述操作还包括:

基于所述配置数据来生成参数值;以及

向所述计算机程序的一部分提供所述参数值,该一部分能够与所述资源进行通信,

所述计算机程序的执行是基于所述参数值的。

22.根据权利要求21所述的数据处理系统,其中,基于所述配置数据来生成参数值的步骤包括:执行用于生成所述计算机程序的所述一部分能够读取的格式的参数文件的参数生成器,以及

向所述计算机程序的所述一部分提供所述参数值的步骤包括:使所述参数文件能够用于所述计算机程序的所述一部分。

23.根据权利要求18所述的数据处理系统,其中,所述SQL查询包括具有自变量的SELECT语句,所述自变量的至少一部分与和所述资源相关联的所述标识符相对应。

24.根据权利要求18所述的数据处理系统,其中,所述计算机程序包括表示所述SQL查询的操作的组件。

25.根据权利要求18所述的数据处理系统,其中,所述计算机程序是数据流图,并且该数据流图的子图进行向所述资源的所述一个或多个指令的发送。

26.根据权利要求18所述的数据处理系统,其中,使用所述资源所公开的应用编程接口即API来访问该资源。

27.根据权利要求26所述的数据处理系统,其中,所述操作还包括:

使得能够响应于接收到所述SQL查询而执行所述API的功能,

所述API的功能能够在所述资源处执行,并且被发送至所述资源的指令使得能够执行所述API的功能。

28.根据权利要求18所述的数据处理系统,其中,所述操作还包括:以数据库表的形式对从外部资源接收到的数据进行格式化。

29.根据权利要求18所述的数据处理系统,其中,所述资源不是关系数据库管理系统。

30.根据权利要求18所述的数据处理系统,其中,所述指令被发送至不会响应于SQL查询而返回结果的资源设施。

31.根据权利要求18所述的数据处理系统,其中,响应于所述指令而从所述资源接收到的数据包括所述SQL查询所指定的数据。

32.根据权利要求18所述的数据处理系统,其中,所述操作还包括:识别响应于所述指令而从所述资源接收到的数据中的记录和字段,所述记录和字段是基于与所述数据处理系统外部的资源相关联的记录格式来识别的。

33. 根据权利要求18所述的数据处理系统,其中,基于所述SQL查询来生成所述计算机程序的步骤包括:

基于所述SQL查询来识别数据源;

识别除管理关系数据库的系统以外的执行系统;

基于所述查询来生成对查询计划器的请求;

向所述查询计划器提供所述请求;

接收所述查询计划器基于所述请求所生成的查询计划,所述查询计划包括对管理关系数据库的系统所要进行的一个或多个步骤的描述;

生成例示包括以下内容的数据流图的数据结构:

第一节点,用于表示要执行的至少一个操作,

所述第一节点与能够被执行系统用来调用可执行程序代码以进行操作的信息相关联,

所述第一节点与能够被执行系统用来使数据能够用于所述程序代码的信息相关联,以

及

所述操作是基于所述查询计划所描述的步骤而选择的,以及

至少一个链接,用于表示所述第一节点的操作的输出数据被提供作为针对所述数据流图的第二节点的操作的输入数据。

34. 根据权利要求18所述的数据处理系统,其中,所述配置数据指定了与所述资源进行通信时所使用的至少一个值。

35. 一种非暂时性计算机可读存储装置,用于存储使得数据处理系统能够基于查询来执行计算机程序的指令,其中,所述查询是根据能够适用于关系数据库的查询语言所表示的,所述计算机程序是至少部分地基于存储在有形、非暂时性计算机可读介质中的数据来执行的,所述指令使得所述数据处理系统进行包括以下内容的操作:

接收SQL查询,其中所述SQL查询包括与所述数据处理系统外部的资源相关联的标识符;

基于所述SQL查询来生成计算机程序,包括:

识别包括与所述资源相关联的至少一个条目的目录,所述至少一个条目包括表示能够用于对计算机程序进行配置以访问所述资源的数据的配置数据的一个或多个数据值,以及从所识别的目录接收所述配置数据;

基于所述配置数据来执行所述计算机程序,所述计算机程序的执行使得:

向所述资源发送一个或多个指令,所述指令定义了除所述SQL查询的操作以外的操作,以及

响应于所述指令而从所述资源接收数据。

36. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述计算机程序包括定义了访问所述资源的方式的可执行指令,所述可执行指令基于被提供至所述计算机程序的配置数据而操作。

37. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,在能够基于所述资源所使用的数据格式的变化而更新的目录中指定所述配置数据。

38. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述操作还包括:

基于所述配置数据来生成参数值;以及

向所述计算机程序的一部分提供所述参数值,该一部分能够与所述资源进行通信,所述计算机程序的执行是基于所述参数值的。

39. 根据权利要求38所述的非暂时性计算机可读存储装置,其中,基于所述配置数据来生成参数值的步骤包括:执行用于生成所述计算机程序的所述一部分能够读取的格式的参数文件的参数生成器,以及

向所述计算机程序的所述一部分提供所述参数值的步骤包括:使所述参数文件能够用于所述计算机程序的所述一部分。

40. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述SQL查询包括具有自变量的SELECT语句,所述自变量的至少一部分与和所述资源相关联的所述标识符相对应。

41. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述计算机程序包括表示所述SQL查询的操作的组件。

42. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述计算机程序是数据流图,并且该数据流图的子图进行向所述资源的所述一个或多个指令的发送。

43. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,使用所述资源所公开的应用编程接口即API来访问该资源。

44. 根据权利要求43所述的非暂时性计算机可读存储装置,其中,所述操作还包括:使得能够响应于接收到所述SQL查询而执行所述API的功能,所述API的功能能够在所述资源处执行,并且被发送至所述资源的指令使得能够执行所述API的功能。

45. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述操作还包括:以数据库表的形式对从外部资源接收到的数据进行格式化。

46. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述资源不是关系数据库管理系统。

47. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述指令被发送至不会响应于SQL查询而返回结果的资源设施。

48. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,响应于所述指令而从所述资源接收到的数据包括所述SQL查询所指定的数据。

49. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,所述操作还包括:识别响应于所述指令而从所述资源接收到的数据中的记录和字段,所述记录和字段是基于与所述数据处理系统外部的资源相关联的记录格式来识别的。

50. 根据权利要求35所述的非暂时性计算机可读存储装置,其中,基于所述SQL查询来生成所述计算机程序的步骤包括:

基于所述SQL查询来识别数据源;

识别除管理关系数据库的系统以外的执行系统;

基于所述查询来生成对查询计划器的请求;

向所述查询计划器提供所述请求;

接收所述查询计划器基于所述请求所生成的查询计划,所述查询计划包括对管理关系数据库的系统所要进行的一个或多个步骤的描述;

生成例示包括以下内容的数据流图的数据结构：

第一节点，用于表示要执行的至少一个操作，

所述第一节点与能够被执行系统用来调用可执行程序代码以进行操作的信息相关联，

所述第一节点与能够被执行系统用来使数据能够用于所述程序代码的信息相关联，以及

所述操作是基于所述查询计划所描述的步骤而选择的，以及

至少一个链接，用于表示所述第一节点的操作的输出数据被提供作为针对所述数据流图的第二节点的操作的输入数据。

51. 根据权利要求35所述的非暂时性计算机可读存储装置，所述配置数据指定了与所述资源进行通信时所使用的至少一个值。

52. 一种数据处理系统，其能够基于根据能够适用于关系数据库的查询语言所表示的查询来执行计算机程序，所述计算机程序是至少部分地基于存储在有形、非暂时性计算机可读介质中的数据来执行的，所述数据处理系统包括：

用于接收SQL查询的部件，其中所述SQL查询包括与所述数据处理系统外部的资源相关联的标识符；

用于基于所述SQL查询来生成计算机程序的部件，其中基于所述SQL查询来生成计算机程序包括：

识别包括与所述资源相关联的至少一个条目的目录，所述至少一个条目包括表示能够用于对计算机程序进行配置以访问所述资源的数据的配置数据的一个或多个数据值，以及从所识别的目录接收所述配置数据；

用于基于所述配置数据来执行所述计算机程序的部件，所述计算机程序的执行使得：

向所述资源发送一个或多个指令，所述指令定义了除所述SQL查询的操作以外的操作，以及

响应于所述指令而从所述资源接收数据。

53. 根据权利要求52所述的数据处理系统，其中，所述计算机程序包括定义了访问所述资源的方式的可执行指令，所述可执行指令基于被提供至所述计算机程序的配置数据而操作。

54. 根据权利要求52所述的数据处理系统，其中，在能够基于所述资源所使用的数据格式的变化而更新的目录中指定所述配置数据。

55. 根据权利要求52所述的数据处理系统，其中，还包括：

用于基于所述配置数据来生成参数值的部件；以及

用于向所述计算机程序的一部分提供所述参数值的部件，该一部分能够与所述资源进行通信，

所述计算机程序的执行是基于所述参数值的。

56. 根据权利要求55所述的数据处理系统，其中，用于基于所述配置数据来生成参数值的部件包括用于执行参数生成器的部件，所述参数生成器用于生成所述计算机程序的所述一部分能够读取的格式的参数文件，以及

用于向所述计算机程序的所述一部分提供所述参数值的部件包括用于使所述参数文件能够用于所述计算机程序的所述一部分的部件。

57. 根据权利要求52所述的数据处理系统, 其中, 所述SQL查询包括具有自变量的SELECT语句, 所述自变量的至少一部分与和所述资源相关联的所述标识符相对应。

58. 根据权利要求52所述的数据处理系统, 其中, 所述计算机程序包括表示所述SQL查询的操作的组件。

59. 根据权利要求52所述的数据处理系统, 其中, 所述计算机程序是数据流图, 并且该数据流图的子图进行向所述资源的所述一个或多个指令的发送。

60. 根据权利要求52所述的数据处理系统, 其中, 使用所述资源所公开的应用编程接口即API来访问该资源。

61. 根据权利要求60所述的数据处理系统, 其中, 还包括:

用于使得能够响应于接收到所述SQL查询而执行所述API的功能的部件,

所述API的功能能够在所述资源处执行, 并且被发送至所述资源的指令使得能够执行所述API的功能。

62. 根据权利要求52所述的数据处理系统, 其中, 还包括: 用于以数据库表的形式对从外部资源接收到的数据进行格式化的部件。

63. 根据权利要求52所述的数据处理系统, 其中, 所述资源不是关系数据库管理系统。

64. 根据权利要求52所述的数据处理系统, 其中, 所述指令被发送至不会响应于SQL查询而返回结果的资源设施。

65. 根据权利要求52所述的数据处理系统, 其中, 响应于所述指令而从所述资源接收到的数据包括所述SQL查询所指定的数据。

66. 根据权利要求52所述的数据处理系统, 其中, 还包括: 用于识别响应于所述指令而从所述资源接收到的数据中的记录和字段的部件, 所述记录和字段是基于与所述数据处理系统外部的资源相关联的记录格式来识别的。

67. 根据权利要求52所述的数据处理系统, 其中, 用于基于所述SQL查询来生成所述计算机程序的部件包括:

用于基于所述SQL查询来识别数据源的部件;

用于识别除管理关系数据库的系统以外的执行系统的部件;

用于基于所述查询来生成对查询计划器的请求的部件;

用于向所述查询计划器提供所述请求的部件;

用于接收所述查询计划器基于所述请求所生成的查询计划的部件, 所述查询计划包括对管理关系数据库的系统所要进行的一个或多个步骤的描述;

用于生成例示包括以下内容的数据流图的数据结构的部件:

第一节点, 用于表示要执行的至少一个操作,

所述第一节点与能够被执行系统用来调用可执行程序代码以进行操作的信息相关联,

所述第一节点与能够被执行系统用来使数据能够用于所述程序代码的信息相关联, 以

及

所述操作是基于所述查询计划所描述的步骤而选择的, 以及

至少一个链接, 用于表示所述第一节点的操作的输出数据被提供作为针对所述数据流图的第二节点的操作的输入数据。

68. 根据权利要求52所述的数据处理系统, 其中, 所述配置数据指定了与所述资源进行



通信时所使用的至少一个值。

## 查询网络上的数据源

[0001] 优先权声明

[0002] 本申请要求提交于2015年6月26日的美国专利申请序列14/752,094的优先权,美国专利申请序列14/752,094要求提交于2015年2月18日的美国专利申请序列62/117,588的优先权,其全部内容通过引用而并入于此。

### 背景技术

[0003] 本说明书涉及例如使用SQL或另一类型的查询语言来查询网络上的数据源。

[0004] 对数据源的查询指定要从数据源检索的数据。查询可被提供给数据源(例如,数据库),并且与数据源相关联的数据处理系统(例如,数据库管理系统)可以返回由查询指定的数据。可以使用各种技术来解析查询以识别数据源中的由查询指定的数据。

### 发明内容

[0005] 方面1是一种利用数据处理系统所进行的计算机实现方法,用于基于根据能够适用于关系数据库的查询语言所表示的查询来执行计算机程序,所述计算机程序是至少部分地基于存储在有形、非暂时性计算机可读介质中的数据来执行的,其中,执行所述计算机程序的步骤包括:接收SQL查询,其中所述SQL查询包括与所述数据处理系统外部的资源相关联的标识符;基于所述SQL查询来生成计算机程序,包括:识别与所述资源相关联的配置数据,所述配置数据指定了与所述资源进行通信时所使用的至少一个值;以及基于所述配置数据来执行所述计算机程序,所述计算机程序的执行使得:向所述资源发送一个或多个指令,所述指令定义了除所述SQL查询的操作以外的操作,以及响应于所述指令而从所述资源接收数据。

[0006] 方面2是根据方面1所述的方法,其中,所述计算机程序的一部分包括定义了访问所述资源的方式的可执行指令,所述可执行指令基于被提供至所述计算机程序的所述一部分的配置数据而操作。

[0007] 方面3是根据方面1或2所述的方法,其中,在能够基于所述资源所使用的数据格式的变化而更新的目录中指定所述配置数据。

[0008] 方面4是根据方面1、2或3中任一项所述的方法,还包括:基于所述配置数据来生成参数值;以及向所述计算机程序的一部分提供所述参数值,该一部分能够与所述资源进行通信,所述计算机程序的执行是基于所述参数值的。

[0009] 方面5是根据方面4所述的方法,其中,基于所述配置数据来生成参数值的步骤包括:执行用于生成所述计算机程序的所述一部分能够读取的格式的参数文件的参数生成器,以及向所述计算机程序的所述一部分提供所述参数值的步骤包括:使所述参数文件能够用于所述计算机程序的所述一部分。

[0010] 方面6是根据方面1到5中任一项所述的方法,所述SQL查询包括具有自变量的SELECT语句,所述自变量的至少一部分与和所述资源相关联的所述标识符相对应。

[0011] 方面7是根据方面1到6中任一项所述的方法,其中,所述计算机程序包括表示所述

SQL查询的操作的组件。

[0012] 方面8是根据方面1到6中任一项所述的方法,其中,所述计算机程序是数据流图,并且所述计算机程序的一部分是该数据流图的子图。

[0013] 方面9是根据方面1到8中任一项所述的方法,其中,使用所述资源所公开的应用编程接口即API来访问该资源。

[0014] 方面10是根据方面9所述的方法,还包括:使得能够响应于接收到所述SQL查询而执行所述API的功能,所述API的功能能够在所述资源处执行,并且被发送至所述资源的指令使得能够执行所述API的功能。

[0015] 方面11是根据方面1到10中任一项所述的方法,还包括:以数据库表的形式对从外部资源接收到的数据进行格式化。

[0016] 方面12是根据方面1到11中任一项所述的方法,其中,所述资源不是关系数据库管理系统。

[0017] 方面13是根据方面1到12中任一项所述的方法,其中,所述指令被发送至不会响应于SQL查询而返回结果的资源设施。

[0018] 方面14是根据方面1到13中任一项所述的方法,其中,响应于所述指令而从所述资源接收到的数据包括所述SQL查询所指定的数据。

[0019] 方面15是根据方面1到14中任一项所述的方法,还包括识别响应于所述指令而从所述资源接收到的数据中的记录和字段,所述记录和字段是基于与所述数据处理系统外部的资源相关联的记录格式来识别的。

[0020] 其它方面可以包括相应的设备、系统和计算机可读存储装置。

[0021] 各方面可以包括以下优点中的一个或多个。数据处理系统可以执行引用该数据处理系统外部的资源(例如,因特网上可用的一个资源)的查询,并且不是关系数据库管理系统。

[0022] 即使网络中的数据源可能不对数据库语言作出反应、或者可能不是关系数据库,也可以通过使用数据库编程语言(例如,SQL)有效地使得遍及网络(例如,诸如因特网等的大型网络)分布的数据可访问。这样可以提供可以特别适合于特定的数据源、并因此可以在具有随时间变化的数据源的网络中使用的灵活的搜索模式。反过来,这样可以使得能够(例如,通过按比例增加或减小数据源的数量、或者通过交换或修改数据源)更快地开发分布式网络,同时保持对存储在所开发网络的当前数据源中的数据的访问。

[0023] 根据以下描述和权利要求书,本发明的其它特征和优点将变得显而易见。

## 附图说明

[0024] 图1示出可以处理查询的数据处理系统。

[0025] 图2A~2C示出数据处理系统的元件。

[0026] 图3示出用于执行查询的用户界面。

[0027] 图4A示出数据流图。

[0028] 图4B示出数据流图的子图。

[0029] 图4C示出组件的可执行代码。

[0030] 图5A~5C示出外部资源的目录的内容。

[0031] 图6A~6D是与执行查询相关联的处理的流程图。

### 具体实施方式

[0032] 数据处理系统可以将查询(诸如SQL查询)转换为诸如数据流图等的计算机程序。计算机程序包括被执行时进行与查询所指定的操作等同的操作(例如,数据处理操作)的组件。

[0033] 图1示出可以处理查询102以生成由查询102指定的结果104的数据处理系统100。结果104基于从原始查询102所引用的外部资源108接收到的数据106。数据处理系统100响应于该数据处理系统100基于查询102所生成、并发送至外部资源108的指令110来接收数据106。

[0034] 查询102用于检索由该查询所指定的数据。一种类型的查询102是结构化查询语言(SQL)查询。本说明书将使用SQL查询作为示例,但是此处所描述的技术也可以与诸如多维表达式(MDX)查询等的其它类型的查询一起使用。

[0035] SQL查询(亦称为SQL语句)使用由结构化查询语言(SQL)定义的命令和句法。一般来说,查询是用于指定该查询中所指示的一个或多个数据集中的数据的子集的语句。指定的子集可以由用于处理查询的系统返回给发出了该查询的系统。由查询指定并且响应于查询而返回的数据通常是存储在查询所指示的数据集中的总数据的一部分。SQL查询的示例可以是“SELECT last\_name FROM current\_customers”。该SQL查询包括操作SELECT,该操作指示执行查询的系统根据SELECT操作的自变量来检索数据。在SQL的句法中,自变量是作为诸如数据库表等的数据集的“current\_customers”和作为数据库表的列的“last\_name”。当系统解释查询并执行查询的操作时,系统将响应于查询而返回last\_name列的数据(例如,包含在last\_name列中的数据的每个部分)。在通过引用而并入于此的ISBN 978-0470229064的由Alex Kriegel和Boris Trukhnov在2008年4月7日发表的“SQL Bible,2nd Edition”中详细描述了SQL。

[0036] 响应于查询而返回数据的资源的典型示例是关系数据库。关系数据库是一个或多个数据库表的集合以及管理诸如解释SQL查询、从表读取数据、向表写入数据以及执行其它种类数据处理功能等的数据处理操作的系统。数据库表是排列在a)各自表示记录的行和b)各自表示存储在行中的数据的类别的列中的数据的集合。例如,名为“current\_customers”的数据库表可以具有各自表示业务的当前客户的行,并且可以具有表示数据的类别(诸如客户的名称、客户的地址、客户最近购买的产品等)的列。

[0037] 关系数据库表是一种用于以元组的形式存储数据的数据库表,这些元组各组由与属性相对应的数据元素组成。元组可以采用关系数据库中的行的形式,而属性可以采用关系数据库中的列的形式。因此,各元组包含各自与数据库表的属性之一相对应的数据元素(有时被称为属性值)。此外,多个元组的分组有时被称为关系。

[0038] 关系数据库管理系统(RDBMS)是用于处理涉及创建并修改存储在关系数据库中的数据的指令的系统。RDBMS包括用于解释查询并返回查询所指定的数据的功能。解释查询和返回查询所指定的数据的组合有时称为执行查询。例如,一些RDBMS实现包括引擎,该引擎a)解析SQL查询、b)识别由结构化查询语言定义的操作、c)识别命令的自变量以及d)根据自变量进行(例如,执行)操作。

[0039] 如上所述,SQL查询“SELECT last\_name FROM current\_customers”包括操作SELECT,该操作指示RDBMS根据SELECT操作的自变量来检索数据。自变量是作为RDBMS所管理的数据库表的“current\_customers”和作为数据库表的列的“last\_name”。当RDBMS解释查询并执行查询的操作时,RDBMS将响应于查询而返回last\_name列的数据(例如,包含在last\_name列中的数据的每个部分)。在RDBMS的一些实现中,称为查询计划器的模块将会识别要进行的操作。以这种方式,SQL是适用于关系数据库(例如,适用于RDBMS所保持的数据)的查询语言。

[0040] 外部资源108在数据处理系统100的外部。(图2A~2C中详细示出数据处理系统100的一些组件。)例如,外部资源108可以是使用网络(例如,因特网)来进行通信的设施。数据处理系统100通过向网络发送数据并从网络接收数据来与外部资源108进行通信。外部资源108可以包括诸如数据库表、数据文件或存储在例如有形、非暂时性计算机可读介质的存储介质上的其它数据结构等的数据集。

[0041] 即使查询102所指定的外部资源108不包括RDBMS,数据处理系统100也能够生成查询102所指定的结果104。换句话说,外部资源108无需是RDBMS,或者在其组件中无需包括RDBMS。外部资源108的一些示例将包括RDBMS,而外部资源108的其它示例将不包括RDBMS。

[0042] 在使用中,数据处理系统100基于查询102而生成计算机程序108(有时称为数据处理程序)。例如,可以使用将查询当作输入并产生数据流图作为输出的引擎(例如,形成数据处理系统100的子系统的引擎)来生成计算机程序108。尽管此处使用数据流图作为示例,但计算机程序可以是包括程序代码的任何种类的程序,所述程序代码可被执行用于进行程序代码所表示的指令。

[0043] 以这种方式,诸如图1所示的计算机程序108等的计算机程序在被执行时可例如通过诸如RDBMS等的数据库管理系统来产生与相应查询102的执行相同的输出。因此,可以使用诸如SQL等的查询语言来编写查询102。然而,进行相应数据处理操作的系统(例如,数据处理系统100的子系统)可以执行计算机程序108,以便执行与执行查询102的系统(例如,RDBMS)将执行的操作等同的操作。(通常,RDBMS没有能够执行计算机程序108的功能。)当我们说两个操作等同时,意思是当提供相同的输入数据时,这两个操作产生基本上相同的输出数据。例如,当提供有相同的输入数据时,两个操作可以产生完全相同的输出数据。例如,提供有相同输入数据的两个操作可以产生仅在数据格式化方面不同的输出数据,例如,一个操作可以产生逗号分隔的输出数据,并且等同的操作可以产生其它与逗号分隔的输出数据相同的制表符分隔的输出数据。

[0044] 即使查询102包括对除RDBMS以外的数据源的引用112,数据处理系统100也可以生成该查询102所指定的结果104。例如,查询102可以引用不是RDBMS的外部资源108。因为外部资源108不是RDBMS,所以外部资源108不与用于解释例如SQL查询等的查询的功能相关联。在一些实现中,如果数据处理系统100可以访问外部资源108的数据的记录格式,则该数据处理系统100可以对从外部资源108接收到的数据应用查询。记录格式(有时称为“模式”)是对数据体的组织的描述。

[0045] 外部资源108可以是在数据处理系统100外部的任何数据源。在数据处理系统100的外部是指资源不是数据处理系统100的组件之一。(图2A~2C中详细示出数据处理系统100的一些组件示例。)例如,外部资源108可以是使用网络114(例如,此处表示为“云”的因

特网)来进行通信的设施。数据处理系统100通过向网络114发送数据并从网络114接收数据来与外部资源108进行通信。在一些示例中,外部资源108可以是使用诸如TCP/IP或UDP/IP等的基于因特网的协议来进行通信的网站或另一设施。

[0046] 在一些实现中,外部资源108可以包括网络114不可见的RDBMS。我们的意思是外部资源108可以包括用于存储外部资源108的数据的关系数据库管理系统,但是该关系数据库管理系统不会接受通过网络114的方式达成的诸如SQL查询等的查询,除极限的情况(例如,诸如通过使得系统管理员能够提交查询的系统管理员界面的方式的情况)以外。例如,如果外部资源108是因特网上可用的网站,则该外部资源108可以具有用于存储数据的“后端”关系数据库。在该示例中,关系数据库不接受来自网络浏览器界面、移动应用的SQL查询、或者外部资源108的许多或大部分用户所使用的其它访问技术的SQL查询。作为替代,外部资源108的数据主要通过不包括诸如超文本传输协议(HTTP)请求等的SQL查询或通过应用编程接口(以下描述为API)所提交的指令的技术、或者另一种技术的方式来访问。

[0047] 因为外部资源108不解释查询102,因此数据处理系统100确定要发送至外部资源108的、可以由该外部资源108所解释的指令110。指令110采用除原始查询102的形式以外的形式。例如,如果原始查询102是SQL查询,则指令110不是SQL查询(例如,指令110不包含SQL命令或自变量)。数据处理系统100可以基于查询102并且基于用于描述外部资源108的其它信息来确定要向外部资源108发送什么指令110。在一些实现中,数据处理系统100具有用于解释查询102并且生成由该查询102所指定的结果104的模块。在一些实现中,数据处理系统100生成包括与进行查询102的操作相对应的操作的计算机程序,并且在被执行时向外部资源108发送指令110。

[0048] 图2A~2C示出可以用于执行引用外部资源108的查询102的数据处理系统100的元件。参考图2A,在数据处理系统100接收到查询102的情况下,该查询102被提供给计算机程序生成引擎120。计算机程序生成引擎120生成计算机程序132,该计算机程序108在被执行时进行与查询102相对应的操作。例如,查询102可以是SQL查询,例如包含由结构化查询语言定义的一个或多个命令以及与操作相关联的自变量的查询。在该示例中,计算机程序132包含与进行SQL查询的操作等同的可执行功能。在(例如,利用执行引擎140)执行计算机程序132的情况下,该计算机程序132基于SQL查询中所定义的同自变量来执行。

[0049] 数据处理系统100可以使用不依赖于RDBMS的功能(例如,RDBMS的查询解释功能)进行操作的技术,来进行用于实行查询102(一种处理,其有时被称为执行查询102)的操作。作为替代可以通过执行计算机程序132来进行查询。一旦生成和配置了计算机程序132,则不使用关系数据库的查询解释功能来生成基于查询102的输出。此外,即使查询102中识别出的数据源不是使用查询102形式的查询进行操作的数据库,数据处理系统100也可以执行查询102。例如,外部资源108可以不被配置为接受以SQL形式指定的指令。如果查询102是SQL查询并且引用外部资源108,则数据处理系统100可以接收查询102并且响应于配置和执行计算机程序132的时间来确定应当进行什么操作,以使得计算机程序132的输出与查询102的执行输出等同。

[0050] 以这种方式,SQL查询可以用于从关系数据库系统以外的系统检索数据。因为SQL是用于指定查询的常用语言,因此许多用户知道如何编写SQL查询,并将许多传统系统配置为自动生成SQL查询。此处描述的技术使得用户和传统系统能够编写或生成SQL查询,并且

该SQL查询可以被进行以从没有用于解释SQL查询的功能的外部资源检索数据。此外,可以由数据处理系统100从外部资源检索数据,而不是将数据从外部资源复制到执行SQL查询的关系数据库。

[0051] 在一些实现中,计算机程序132包括数据流图。数据流图是包含表示要对输入数据执行的操作的组件以及组件(有时称为节点)间的表示数据流动的链接的计算机程序。组件所表示的操作通过处理输入数据来生成基于输入数据的输出数据。如果组件链接到其它组件,则该组件可以向该其它组件提供输入数据并从该其它组件接收输出数据,其中,两个组件之间的每个链接表示数据从一个组件向另一个组件的流动。图的组件的子集(例如,图的一个或多个组件)有时称为图的子图。

[0052] 当通过基于图的处理系统来执行数据流图时,执行各个组件(例如,执行计算机程序或计算机程序的一部分)并且进行该组件所表示的操作。在执行期间,数据流图接收处理后(例如,通过数据流图的组件的操作而进行操作)的输入数据以生成输出数据。

[0053] 数据流图的一些或所有组件各自与用于调用可执行程序代码的信息相关联,以执行与该组件相关联的操作。在一些实现中,表示数据流图的数据结构可以包括引用可执行代码的数据。数据结构可以用于例示数据流图,我们的意思是数据结构可以用于执行进行与数据流图相关联的操作的代码。例如,组件可以与包含用于进行与组件相关联的操作(例如,处理和输出数据)的计算机可执行指令的计算机可读存储器中所存储的计算机程序的引用相关联。

[0054] 在一些示例中,数据流图的一些或所有组件各自与用于使数据可用于程序代码的信息相关联。例如,组件可以与可被调用以将数据递送至与组件相关联的可执行程序的函数调用相关联,或者组件可以与可接收被递送至与组件相关联的可执行程序的数据的网络端口相关联,或者组件可以与用于将数据递送至与组件相关联的可执行程序的另一种技术相关联。以这种方式,每个组件可以接收、处理并输出数据。

[0055] 在一些示例中,数据流图是可参数化的,我们的意思是在准备供执行的数据流图时,可以使用参数的值来配置该数据流图。提供了参数值的数据流图的实例有时被称为数据流图的参数化实例。参数是可以变化以改变参数所属程序的行为的一种数据类型。例如,可以向程序提供参数的值,以便改变程序处理输入数据以产生输出数据的方式。在数据流图的情况下,数据流图的各组件可以与一个或多个参数相关联。类似地,一个或多个参数可以与单个组件或多个组件相关联。

[0056] 在通过引用而并入于此的标题为“Managing Parameters for Graph-Based Applications”的美国公布2007/0011668中详细描述了基于图的系统的一个示例。在通过引用而并入于此的标题为“Executing Computations Expressed as Graphs”的美国专利5,966,072中描述了一种用于执行基于图的计算的系统。此外,数据流图的组件可以替代查询102的操作。在通过引用而并入于此的标题为“Managing Data Queries”的美国公布2011/0179014A1以及标题也为“Managing Data Queries”的美国公布2012/0284255A1中进一步描述了根据这种替代的技术。在一些实现中,数据流图可以由查询102产生。

[0057] 参考图2B,在一些示例中,计算机程序132包括在被执行时与外部资源108进行通信的部分134。例如,该部分134可以包括被配置为向外部资源108发送数据以及/或者从外部资源108接收数据的可执行功能(例如,可执行程序代码)。在执行该部分134(例如,利用

执行引擎来解释和进行该部分134的程序代码)时,计算机程序134向外部资源108发送数据以及/或者从外部资源108接收数据。在一些实现中,该部分134(以及计算机程序132的其它部分)可以包括被提供给数据处理系统100的程序代码,并且基于外部资源108的身份来进行检索。例如,程序开发者(未示出)可能已经编写了该部分134的程序代码以符合外部资源的技术要求。在一些实现中,技术要求包括下述的应用编程接口(API)。在一些实现中,该部分134是数据流图的组件或数据流图的子图。

[0058] 在一些实现中,计算机程序生成引擎120通过访问计算机程序的一个或多个预先存在的部分并组装这些部分以形成计算机程序132,来生成计算机程序132。例如,计算机程序生成引擎120可以访问用于存储预先存在的组件(例如,可执行程序代码的预先存在的部分)的组件库126。例如,预先存在的组件可以是适于包含在数据流图中的组件,或者可以是适于包含在数据流图中的另一个图(例如,子图)。

[0059] 在一些实现中,组件库126可以包括各自与特定种类的外部资源108相对应的预先存在的组件。例如,组件库126可以包括与查询102中的通过引用112所表示的外部资源108相对应的组件124。在一些示例中,开发者为了使得数据处理系统100能够访问与引用112相对应的外部资源108的目的而可能已经开发了组件124。可以基于外部资源108的身份从组件库126中选择组件124。

[0060] 当计算机程序生成引擎120(例如,从组件库126)接收到一个或多个组件124时,该计算机程序生成引擎120还配置了一个组件124或多个组件。这样做时,计算机程序生成引擎120生成具有配置部分的计算机程序132。例如,所配置计算机程序132包括与组件124或者包括用于与外部资源108进行通信的可执行功能的多个组件相对应的配置部分134。

[0061] 在一些实现中,可以基于计算机程序生成引擎120接收到的性质136来对配置部分134进行配置。性质136包括用于描述外部资源108的特征的数据。计算机程序132使用该数据来确定如何发送数据并且从外部资源108接收数据。在一些示例中,外部资源108可以以特定格式发送和接收数据。在这些示例中,性质136可以包括要被发送以及从外部资源108接收到的数据的格式规范。在一些示例中,外部资源108可能需要使用诸如用户名和/或密码等的凭证。在这些示例中,性质136可以包括标识符的规范。在一些示例中,外部资源108可以在特定地址或其它位置处可访问。例如,外部资源108可以在特定IP(因特网协议)地址或特定服务器名称或另一类型的地址处可访问。在这些示例中,性质136可以包括地址信息。

[0062] 可以从可例如响应于外部资源108的操作变化而更新的源接收性质136。例如,性质136可以存储在目录122中,其中该目录122包含表示与数据处理系统100能够进行通信的外部资源108相对应的性质的数据。目录122可以是数据库、平面文件或任何其它类型的数据存储机构。在一些实现中,目录122可以从多个实体同时读取和写入。例如,目录122可以被实现为具有用于管理并发数据读写操作的功能的数据库或其它数据存储技术。用于管理并发数据读写操作的功能的一个例子是用于指示何时写入一部分数据的锁或信号量的使用。可以使用的一种类型的锁定功能是两相锁定,其中获取一个或多个锁,进行读写操作,然后释放锁。通过管理并发数据读写操作,生成引擎120的许多实例可以使用单一目录122。此外,单一目录122可以存储许多类型的外部资源108的性质136。即使目录122所支持的外部资源108的数量足够大以使得性质136经常被访问(例如,使得每秒读取或写入存储在目



录136中的至少一些性质数次或更多次),与任何外部资源108相关联的性质136也可以在任何时间更新。

[0063] 在一些示例中,如果外部资源108的操作发生改变,则用于与外部资源108进行通信的任何可执行代码(例如,组成可执行组件124的一部分的可执行代码)无需改变;只更新性质。例如,外部资源108可以改变其向其它系统提供输出数据的方式。性质136使得计算机程序生成引擎120能够以配置部分134将被配置为接受将从外部资源108接收到的格式的數據的方式,来对所配置计算机程序132的一部分134进行配置。此外,所配置计算机程序132可以生成要被发送至外部资源的指令110,并且在被外部资源108解释时将会使该外部资源108响应于指令110而发送回数据106。指令110可以由配置部分134基于性质136所指定的指令的格式来格式化。在一些实现中,指令的格式是从存储在目录122中的数据所推导出的。

[0064] 在一些实现中,目录122包括与数据处理系统100能够进行通信的外部资源108的数据106的记录格式有关的信息。例如,计算机程序132可以使用存储在目录122中的记录格式128来解释从相应的外部资源108接收到的数据106。在一些示例中,存储在目录122中的记录格式128指定从外部资源108接收的数据106的结构。记录格式可以指定数据被组织为记录的数据结构,使得各记录具有多个字段。计算机程序132可以使用记录格式128来识别从外部资源108接收到的数据106中的记录和字段。在一些示例中,计算机程序132可以使用记录格式128来解释数据106,例如,将数据的记录和字段转换成不同的格式(例如,计算机程序132的输出中所要使用的格式)。

[0065] 在一些实现中,计算机程序132可以使用目录122的记录格式128来以与SQL查询的结果的预期形式相对应的形式输出数据。在一些示例中,关系数据库管理系统例如响应于SQL查询以记录和字段的形式返回数据。因此,如果计算机程序132还可以识别来自外部资源108的数据106中的记录和字段,则计算机程序132可以响应于SQL查询以记录和字段的形式提供输出。即使从外部资源108接收到的数据106没有采用通常从关系数据库(例如数据库表)接收到的数据的形式,也可以执行这一点。

[0066] 在一些实现中,计算机程序132可以配置有参数。例如,参数可以是可被改变以改变程序行为的值。作为具体示例,参数可以是“filename”并且该参数的值可以是文件在文件系统中的位置。参数的值可以改变为不同文件的位置,以将程序配置为访问不同的文件。同一程序的两个实例(例如,同一可执行程序代码的实例)可以配置有不同的参数值,这将改变同一程序的两个实例的行为。

[0067] 参考图2C,计算机程序生成引擎120可以使用性质来生成用于对所配置的计算机程序132的一部分134进行配置的参数值138。在一些示例中,计算机程序132可以具有与各性质136相对应的参数。例如,计算机程序132可以具有称为“record\_format”的参数,用于确定计算机程序132将接收数据106的格式。计算机程序生成引擎120可以基于作为外部资源108的性质136的一部分而接收的记录格式,来生成针对“record\_format”参数的参数值138。作为另一示例,计算机程序132可以具有称为“用户名”的参数,用于在连接至外部资源108时提供用户名。计算机程序生成引擎120可以基于作为外部资源108的性质136的一部分而接收的用户名数据,来生成针对“用户名”参数的参数值138。

[0068] 在一些实现中,参数值138以计算机程序132可读取的格式的参数文件的形式被提供给计算机程序132。例如,参数文件可以以数据操作语言(DML)格式化。

[0069] 在一些实现中,基于与外部资源108相关联的可执行代码来生成计算机程序132。例如,目录122可以指定可执行代码的位置,以使得当可执行代码(例如,使用诸如性质136等的配置数据)进行配置时,该可执行代码可以用于与外部资源108进行通信。以这种方式,如果例如由被授权改变可执行代码的系统管理员或其它实体更新可执行代码,则目录122存储更新后的可执行代码的位置。当计算机程序生成引擎120生成计算机程序132时,该计算机程序生成引擎120可以访问目录以确定可执行代码的最新版本的位置。

[0070] 一旦生成计算机程序132,执行引擎140就接收该计算机程序132。然后,执行引擎140执行计算机程序132,例如进行与计算机程序132相关联的程序代码所指定的指令。当执行时,计算机程序132(例如,计算机程序的配置部分134)生成指令110并将指令110发送至外部资源108。在一些实现中,外部资源108公开用于向外部资源108发送指令并从外部资源108接收数据的API 142(应用编程接口)。一般来说,API 142可以是使得计算机程序132能够与外部资源108进行交互的任何设施。例如,API 142可以指定外部资源108被配置为接收和执行的指令的类型。API通常所指定的指令的示例是函数调用。函数是可执行程序代码的一部分。在使用函数调用的情况下,计算机程序132发送作为指令110的一部分的函数名称以及要被传递至该函数的(例如,被函数的可执行程序代码所使用的)自变量。指令110可以包括许多函数调用、或其它类型的指令、或这两者。

[0071] 在一些实现中,如果计算机程序132是数据流图,则执行引擎140包括专门的操作系统,有时称为图操作系统。图操作系统是能够执行数据流图的各个组件背后的操作的计算机程序。例如,如果数据流图的组件表示数据处理系统要进行的操作,则图操作系统的任务是指示数据处理系统进行该操作。

[0072] 在将指令110发送至外部资源108之后,计算机程序132响应于指令110从外部资源接收数据106。然后执行引擎140将所接收到的数据106格式化为查询102的结果104。以这种方式,数据处理系统100可以执行查询102以生成由该查询指定的结果104。

[0073] 在一些实现中,外部资源108提供诸如以属性值对的形式指定数据的JSON(JavaScript对象表示法)、或者指定通过表示数据类别的标签所标定的数据的XML(可扩展标记语言)等格式的数据106。在一些实现中,数据106的记录格式128与和外部资源108相关联的其它数据相关联地存储在目录122中,并且可以用于解释JSON或XML数据。

[0074] 当计算机程序132由执行引擎140执行时,计算机程序132(例如,结果104)与用于执行查询102但不执行所配置计算机程序132的(除执行引擎140以外的)系统的输出等同。以这种方式,计算机程序132是与查询102相对应的计算机程序的示例。在一些实现中,计算机程序132以数据库表144的形式对结果104进行格式化。以这种方式,数据处理系统100可以执行查询102(例如,诸如SQL查询等的数据库查询),并且提供与在诸如RDBMS等的系统执行查询102并且提供查询所指定的结果的情况下将会提供的相同种类的输出。

[0075] 此外,虽然SQL查询通常用于检索存储在数据库表中的数据,但是此处所示的查询102可以用于检索以若干可能形式中的任意形式存储的数据。从外部资源108接收到的数据106可以是除数据库表(例如,关系数据库表)以外的形式,例如平面文件。平面文件是不包含数据元素之间的结构关系的数据文件。平面文件的数据可能能够以记录和字段的形式表示。在一些示例中,从外部资源108接收到的数据106可以是外部资源108的原始格式,我们的意思是数据106可以是外部资源108用来存储和处理数据的格式。

[0076] 图3示出用于执行查询的用户界面300。该用户界面300与图1和2所示的数据处理系统100相关联。该用户界面300使得用户(未示出)能够在用户界面300的文本框304中输入SQL查询302。SQL查询302包括作为数据源的标识符的“SELECT”命令306和自变量308。此处,自变量308是“web.wikipedia”。自变量“web.wikipedia”与外部资源(例如,图1所示的外部资源108)相关联。

[0077] 在一些示例中,数据处理系统100可以存储标识符的列表以及与各标识符相关联的外部资源的列表。可以使用许多技术来将标识符与外部资源相关联。在一些示例中,参考图2B~2C,目录122可以存储各自与外部资源相对应的标识符的列表。例如,各标识符可以与被提交至数据处理系统100的查询102中所包括的引用112相对应。例如,引用112可以是图3中的查询300所示的自变量308“web.wikipedia”。以这种方式,可以从查询102中提取引用112,并且使用该引用112来在目录122中查找相应的外部资源。此外,目录可以将各标识符与相应的一组性质136相关联。目录122可以将各标识符与对组件库126中的一个或多个组件的引用相关联。以这种方式,可以使用对外部资源的引用来在目录122中识别相应的一组性质以及对相应组件(例如,如图2B~2C所示的组件124)的引用。作为另一示例,组件库126可以存储标识符,例如,可以存储各组件124的标识符,使得可以使用标识符(诸如外部资源的名称)来识别一个或多个相应组件(例如,如图2B~2C所示的组件124)。

[0078] 如图3所示,“web.wikipedia”是经由超文本传输协议(HTTP)来引用因特网上可用的称为WIKIPEDIA的网站的标识符。当用户单击运行查询按钮310时,用户界面300使数据处理系统100(图1)执行SQL查询302。

[0079] 一旦执行了SQL查询302,就在用户界面300中示出执行的结果312。在该示例中,结果312以数据库表的形式显示。结果312包括包含数据元素的行314和列316。尽管与自变量308相关联的外部资源不是关系数据库,但数据处理系统100使得能够使用外部资源(此处为WIKIPEDIA)作为数据源来执行查询302,并且使得结果312能够被格式化为数据库表。例如,可以通过识别WIKIPEDIA所返回的数据中的记录和字段来将结果312格式化为数据库表。可以通过使用与WIKIPEDIA相关联的记录格式128(图2B~2C)来进行识别。此处,查询302包括用于表示结果312必须满足的标准的“where”命令318。命令318具有用于表示结果必须在文本中包括术语“SQL”的自变量320“subject=‘SQL’”(这种类型的自变量有时被称为谓词)。因此,结果312都包括术语“SQL”。

[0080] 图4A示出响应于查询302的执行而生成的数据流图400。数据流图400是图2A~2C所示的所配置计算机程序132的示例。在一些实现中,数据流图400可以是包含其它组件的另一数据流图的子图。当执行查询302时,数据处理系统(图1和2)生成并配置数据流图400。此外,执行引擎140(图2A~2C)执行数据流图400以生成图3所示的结果312。数据流图400包括表示外部资源108(图1)的组件402和表示针对外部资源108的输入的组件404。此处,组件402表示WIKIPEDIA。例如,组件402、404可以一起是图2B~2C所示的配置部分134的示例。当(例如,利用图操作系统)执行数据流图400时,组件402向外部资源108发送指令(例如,图1和2所示的指令110)。这些指令基于从输入组件404接收到的输入数据。例如,WIKIPEDIA可以公开搜索查询功能,该搜索查询功能执行经由超文本传输协议(HTTP)提交的纯文本查询。在该示例中,组件402被配置为向WIKIPEDIA发送纯文本搜索查询。例如,SQL查询302包括自变量320“subject=‘SQL’”。此处,输入组件404可以被配置为向与WIKIPEDIA进行通信

的组件402提供术语“SQL”。反过来,组件402被配置为发送作为纯文本搜索查询的一部分的(例如,从输入组件404接收到的)术语“SQL”。响应于纯文本搜索查询,组件402经由HTTP来接收结果,其中HTTP可以根据超文本标记语言(HTML)进行格式化。组件402还被配置为解析所接收到的HTML数据并将解析后的数据格式化为诸如数据库表等的形式,例如图3所示的结果312。

[0081] 为了与WIKIPEDIA进行通信,使用WIKIPEDIA特有的性质(例如,图2B~2C所示的性质136)来对组件402、404进行配置。例如,性质136可以包括用于描述WIKIPEDIA提供搜索结果的格式(例如,搜索结果中使用的特定HTML格式)的信息。以这种方式,如果WIKIPEDIA改变其输出的格式,则可以更新(例如,通过目录122的管理员来更新)(例如,存储在图2B~2C所示的目录122中的)性质136,以反映更新后的格式。在对与WIKIPEDIA进行通信的组件402进行配置时,可以访问和使用性质136的最新版本。

[0082] 组件402、404设置有基于性质的参数值。作为示例,组件404可以具有称为“input\_text”的参数,因此组件404可以被配置为使用文本字符串“SQL”作为“input\_text”参数的值。以这种方式,在查询WIKIPEDIA时,使用由SQL查询中定义的标准320所定义的标准。

[0083] 在一些实现中,与诸如WIKIPEDIA等的外部资源进行通信的组件402由多个可执行组件组成。在一些示例中,组件402可以包括具备用于建立与一个或多个网络资源的网络连接的可执行指令的一个或多个组件。例如,网络资源可以是可以提供源自WIKIPEDIA的数据的服务器。

[0084] 在一些示例中,组件402可以包括具备用于解析从外部资源接收到的数据的可执行指令的一个或多个组件。例如,从WIKIPEDIA接收到的数据可以包括响应于SQL查询302的数据和其它种类的数据,诸如标记数据(例如,诸如XML标签的标签)、诸如用于描述所接收到的数据的大小或字符集的数据等的元数据、或者不响应于查询的其它数据。包括用于解析从外部资源接收到的数据的可执行指令的组件可以处理从外部资源接收到的数据,以将响应于查询的数据和不响应于查询的数据分离开。不响应于查询的数据可以被丢弃或用于其它目的。例如,可以使用用于描述所接收到的数据的大小的数据来判断为已经从外部资源接收到预期数量的数据。

[0085] 在一些示例中,组件402可以包括具备用于对从外部资源接收到的数据进行格式化的可执行指令的一个或多个组件。例如,组件402可以(例如,向数据流图400的其它部分)提供以特定方式进行格式化的输出数据。输出数据可以例如被格式化为多行文本、或多元素数组、或其它类型的格式。在一些实现中,包括用于对从外部资源接收到的数据进行格式化的可执行指令的组件可以接收解析后的数据(例如,由包括用于解析从外部资源接收到的数据的可执行指令的这些组件所解析的数据),并且将解析后的数据格式化为被指定用于数据流图400的输出数据的格式。例如,被指定用于数据流图400的输出数据的格式可以是数据库表的格式。

[0086] 由多个组件组成的一种类型的组件称为子图。图4B示出包含组成用于与WIKIPEDIA进行通信的组件402的组件412a~g的子图410的示例。组件412a~g各自包括用于进行与WIKIPEDIA通信所需的计算操作的一部分的可执行功能。例如,一个组件412a是包括用于与网络上的服务器(例如,使用超文本传输协议(HTTP)的服务器)进行通信的可执行代码的“呼叫Web服务”组件。“呼叫Web服务”组件412a向服务器发送请求,并且响应于请求

而接收数据。在一些实现中，“呼叫Web服务”组件412a向外部资源发送简单对象访问协议(SOAP)请求。在一些实现中，“呼叫Web服务”组件412a访问外部资源的应用编程接口(API)。

[0087] “呼叫Web服务”组件412a接收用于指定要在请求中发送的数据的输入。

[0088] 复制组件412b和重新格式化组件412c准备要由“呼叫Web服务”组件412a发送的数据。例如，复制组件412b可以从图4A所示的输入组件404接收数据。在一些实现中，重新格式化组件412c然后可以移除不适于发送到外部资源的数据。例如，一些数据可能不适于发送到外部资源，例如，一些数据可能是可被发送到外部资源的搜索功能的搜索项。在一些示例中，一些数据可能是不适于发送到外部资源的数据。例如，外部资源可能缺少用于以响应于原始查询(例如，图1所示的查询102)的方式来处理数据的功能。作为示例，一些数据可以是正则表达式，并且外部资源可能缺少用于评估外部资源的搜索功能所接收的正则表达式的功能。

[0089] 第二重新格式化组件412d改变从外部资源接收的数据的格式，例如，解析并格式化由“呼叫Web服务”组件412a的输出所提供的数据。例如，从“呼叫Web服务”组件412a接收到的数据可以是诸如XML数据等的标签数据。重新格式化组件412d可以提取标签所标定的数据，并且以不使用XML标签的格式输出所提取的数据。重新格式化组件412d还可以移除输出数据中不需要的数据。例如，丢弃的数据可以是XML标签、或者由不响应于查询的XML标签所标定的数据。在一些实现中，重新格式化组件412d使用记录格式128(图2B~2C)在对从“呼叫Web服务”组件412a接收到的数据进行重新格式化之前解释该数据的结构，例如，识别该数据中的记录和字段。

[0090] 复制组件412b还可以向第三重新格式化组件412e提供数据。在一些实现中，该重新格式化组件412e识别不适于发送至被第一重新格式化组件412c移除的外部资源的数据。此处，不适于发送至外部资源的数据可以由重新格式化组件412e提供给用于组合从两个组件接收到的数据的结合组件412f。

[0091] 第二重新格式化组件412d也将其输出提供给结合组件412f。因此，结合组件412f提供表示来自第二重新格式化组件412d和第三重新格式化组件412e的输入的输出。

[0092] 过滤组件412g接收结合组件412f的输出。过滤组件412g移除不响应于查询(例如，图1所示的查询102)的数据。作为示例，“呼叫Web服务”组件412a可以输出不响应于原始查询的数据，这是因为“呼叫Web服务”组件412a未被提供查询的一个或多个元素(例如，部分)。查询的一个或多个元素可以是正则表达式，其可能不适合于提供给“呼叫Web服务”组件412a所表示的外部资源。此处，过滤组件412g可以接收(例如，由第三重新格式化组件412e输出到结合组件412f、并且由第一重新格式化组件412c从被提供给“呼叫Web服务”组件412a的输出中移除的)正则表达式。过滤组件412g可以基于该正则表达式来过滤从“呼叫Web服务”组件412a首先接收到的数据。尽管此处使用正则表达式的示例，但过滤组件412g可以基于其它类型的查询元素或者其它类型的标准(例如，查询中未指定的标准)来过滤数据。

[0093] 图4B所示的子图410中的各组件具有其自己的、在执行子图410时所进行的可执行代码。例如，图4C示出组成图4B所示的呼叫Web服务组件412的可执行代码440的示例。可执行代码440包括用于将向外部资源的请求进行格式化的代码442、用于解析来自外部资源的输出的代码444、用于向外部资源发送请求的代码446、以及用于将来自外部资源的输出提

供至其它组件的代码448。

[0094] 如以上关于图2B~2C所述,目录122包含与数据处理系统100能够进行通信的外部资源108相对应的数据。存储在目录108中的数据(例如,性质136)用于对计算机程序132进行配置。

[0095] 如图5A所示,在一些实现中,用户界面500可以用于查看和编辑存储在目录122中的数据。用户界面500使得用户(例如,目录122的管理员)能够查看并编辑目录中的条目502的数据。目录的条目502存储与特定外部资源(例如,WIKIPEDIA)相关的数据。尽管此处示出用户界面500和条目502的一个示例,但其它类型的目录可以具有其它类型的条目,并且其它类型的条目可以包含不同种类的数据。

[0096] 该条目502包括与条目有关的诸如一般信息504a等的的数据。条目502还可以包括与条目502相关联的外部资源的记录格式504b,其中该记录格式504b指定从外部资源接收到的数据的格式,如下所述。条目502还可以包括针对该条目的索引504c,其中该索引504c指定可以用于查询与该条目502相关联的外部资源的数据的名称,如下所述。例如,条目502还可以包括针对该条目的关键字504d。关键字用于以记录的形式存储数据的数据源,诸如关系数据库。关键字是一种属性,针对该属性的记录存储值是唯一的。条目502还可以包括针对该条目的参数504e。在一些实现中,参考图2C,参数(例如,参数138)用于对计算机程序132进行配置。在一些实现中,参数504e包括图2B~2C所示的性质136的数据值,其包括用于描述与该条目502相关联的外部资源的特征的数据。例如,参数504e可以包括配置数据,诸如用于获得对外部资源108的授权访问的凭证(例如,用户名和密码)。条目还可以包括针对该条目502的权限504f,其指示出哪些实体(例如,诸如目录122的管理员等的用户界面500的哪些用户)可以访问和/或修改条目502。条目还可以包括针对该条目502的统计504g,其指定与和该条目502相关联的外部资源108有关的统计数据,例如,诸如在外部资源108处可用的数据的数量等的统计数据。

[0097] 如图5A所示,条目的一般信息504a包括对所存储的可执行代码的路径的引用506。在一些实现中,参考图1,所存储的可执行代码是用于与外部资源108进行通信的计算机程序132的一部分134。生成引用120可以使用引用506,以访问所存储的可执行代码并使用所存储的可执行代码来生成计算机程序132。

[0098] 如图5B所示,条目502还可以包括与条目502相关联的外部资源的记录格式504b。记录格式504b可以是图2B~2C所示的记录格式128的示例。记录格式504b指定从外部资源接收的数据的格式。为了检查,记录格式504b可以包括嵌入记录格式508和字段510。字段510指定被发送至外部资源并从外部资源接收的数据的名称。例如,此处,字段510是作为被发送至WIKIPEDIA的数据的名称的“主题”(例如,在对百科全书的查询中所使用的主题)、以及作为从WIKIPEDIA接收到的数据的名称的“行”(例如,与主题相对应的百科全书条目的行)。嵌入记录格式508指定与被提供给外部资源并从外部资源接收的数据有关的格式信息。例如,该嵌入记录格式508表示使用UTF-8来对“主题”和“行”字段进行格式化,其中UTF-8是特定种类的Unicode字符编码。

[0099] 如图5C所示,条目502还可以包括针对该条目的索引504c,其中该索引504c指定可以用于查询与该条目502相关联的外部资源的数据的名称。例如,该条目502具有称为“主题”的索引514。因此,外部资源被配置为接收“主题”的数据值并且响应于该数据值来返回

数据。例如, WIKIPEDIA可以被配置为返回数据值出现在与各条目相关联的“主题”字段中的百科全书条目。

[0100] 图6A示出表示用于执行查询的过程600的流程图。过程600可例如通过图1所示的数据处理系统100的组件来进行。

[0101] 该过程接收 (602) SQL查询。例如, SQL查询可以是图1所示的查询102的示例。SQL查询包括与数据处理系统外部的资源相关联的标识符。此外, 该资源不是关系数据库管理系统。该资源可以是图1所示的外部资源108。标识符的示例是图3所示的自变量308。

[0102] 过程基于SQL查询来生成 (604) 计算机程序。该计算机程序可以是图2A~2C所示的计算机程序132。在一些示例中, 计算机程序包括表示SQL查询的操作的组件。在一些示例中, 计算机程序是数据流图, 并且与外部资源进行通信的计算机程序的一部分是数据流图的组件或子图。图4A中示出数据流图400的示例。在一些实现中, 可以在计算机程序的生成期间进行图6B所示的过程, 以例如在计算机程序的生成期间对该计算机程序的一个或多个部分进行配置。在一些实现中, 可以进行图6D所示的过程以生成可以用于例示计算机程序的数据结构, 例如, 数据流图。

[0103] 该过程例如基于用于对计算机程序进行配置的参数值来执行 (606) 该计算机程序。例如, 计算机程序可以由图2A~2C所示的执行引擎140来执行。在一些实现中, 计算机程序的执行可以与图5C所示的过程620相对应。

[0104] 图6B示出表示用于在生成计算机程序时对该计算机程序的一部分进行配置的过程610的流程图。在一些实现中, 过程610表示用于生成 (604) 如图6A所示的计算机程序的步骤。过程610可例如通过图1所示的数据处理系统100的组件 (诸如计算机程序生成引擎120) 来进行。在一些实现中, 通过组装多个部分来生成计算机程序。当组装计算机程序时, 可以对各部分进行配置。该过程610可以用于对正组装的计算机程序的一部分进行配置。

[0105] 该过程识别 (614) 与外部资源相关联的配置数据。例如, 外部资源可以是图1和2所示的外部资源108。配置数据指定计算机程序用来访问外部资源并从外部资源接收数据的信息。在一些实现中, 配置数据指定要从资源接收的数据的格式, 例如, 图2B~2C所示的记录格式128。在一些实现中, 配置数据指定用于访问网络上的资源的数据。例如, 配置数据可以包含用于获得对外部资源的数据的授权访问的诸如用户名和密码等的凭证, 以及/或者配置数据可以包含诸如主机名或统一资源定位符 (URL) 等的网络地址, 以及/或者配置数据可以包含其它种类的数据。例如, 配置数据可以是图2B~2C所示的性质136的示例。在一些示例中, 在能够基于资源所使用的数据格式的变化而更新的目录中指定配置数据。

[0106] 该过程基于配置数据来生成 (616) 参数值。例如, 参数值可以是图2C所示的参数值138。在一些实现中, 参数生成器接收数据并且生成计算机程序可读取的格式的参数文件。然后将该参数文件提供给计算机程序。例如, 参数生成器可以是图1所示的计算机程序生成引擎120的一部分。参数生成器所接收到的数据可以是图2B~2C所示的性质136的数据。例如, 参数生成器可以解析性质136以确定参数值138。

[0107] 该过程向计算机程序的一部分提供 (516) 参数值。该部分能够与资源进行通信。例如, 计算机程序的该部分可以是图2B~2C所示的配置部分134。计算机程序的该部分包括用于定义访问资源的方式的可执行指令。此外, 可执行指令基于被提供给计算机程序的该部分的参数而操作。



[0108] 可以针对正生成的计算机程序的任意部分来进行该过程610。一旦计算机程序的所有部分都已配置完成,计算机程序就可以执行了。

[0109] 图6C示出表示用于执行与查询相对应的计算机程序的过程620的流程图。在一些实现中,该过程表示用于执行(606)如图6A所示的计算机程序的步骤。该过程620可例如通过图1所示的数据处理系统100的组件(诸如计算机程序执行引擎140)来进行。该计算机程序可以是图2A~2C所示的计算机程序132。例如,当计算机程序执行引擎140接收计算机程序132并且进行与该计算机程序132的可执行代码相对应的操作时,可以启动过程620。

[0110] 计算机程序的执行使一个或多个指令被发送(622)至外部资源。例如,外部资源可以是图1和2所示的外部资源108。指令定义除与计算机程序相对应的SQL查询的操作以外的操作。例如,指令可以是图1所示的指令110。在一些示例中,指令被发送至不会响应于SQL查询而返回结果的外部资源的设施,例如,指令不会被发送至接收SQL查询作为输入、评价SQL查询并且对SQL查询所指定的数据作出响应的外部资源的设施。以这种方式,在一些示例中,指令不会包括诸如SQL命令或自变量等的SQL指令。在一些实现中,使用资源所公开的应用编程接口(API)来访问该资源。图2C中示出API 142的示例。

[0111] 计算机程序的执行使得能够响应于指令而从资源接收(624)数据。例如,数据可以是图1所示的结果104。结果可以以数据库表(例如,图1所示的数据库表144)的形式格式化。响应于指令而从资源接收到的数据包括SQL查询所指定的数据。以这种方式,如果向资源提供SQL查询,并且作为响应,外部资源返回该SQL查询所指定的数据,则接收到的数据与本来会从外部资源接收的数据相同。

[0112] 图6D示出表示用于在例示与查询相对应的数据流图的计算机中生成数据结构的过程630的流程图。例如,数据流图可以是图2A~2C所示的计算机程序132。图4A中示出与查询相对应的数据流图400的示例。

[0113] 生成(632)对基于查询的查询计划器的请求。查询计划器是用于生成表示可被进行以执行查询的步骤的数据的系统。查询计划器通常以RDBMS可以进行的步骤的形式生成数据。

[0114] 接收(634)查询计划器基于请求所生成的查询计划。查询计划是用于表示管理关系数据库的系统(例如,RDBMS)所要进行的一个或多个步骤的描述的数据。

[0115] 例如基于查询来识别(636)数据源。数据源可以是外部资源,例如,图1和2所示的外部资源108。可以基于查询中所表示的引用(例如,图1所示的引用112)来进行识别。在一些示例中,基于查询中的自变量来识别数据源。

[0116] 识别(638)除管理关系数据库的系统以外的执行系统。执行系统可以是用于执行数据流图的系统。例如,执行系统可以是图2A~2C所示的执行引擎140。相比之下,管理关系数据库的系统的示例是RDBMS。

[0117] 生成(640)例示数据流图的数据结构。如以上关于图2A~2C所述,数据流图是包含表示要对输入数据执行的操作的组件以及组件间的表示数据流动的链接(有时称为节点)的计算机程序。

[0118] 数据结构包括用于表示要执行的至少一个操作的节点。节点与执行系统可以用来调用可执行程序代码进行的操作的信息相关联。节点还与执行系统可以用来使数据可用于程序代码的信息相关联。数据流图中的至少一个链接表示该节点的操作的输出数据被提供作



为数据流图中的另一节点的操作的输入数据。

[0119] 基于查询计划所述的步骤来选择操作。在一些实现中,操作包括访问数据源处可用的数据。例如,图4A所示的图400中的组件402是用于访问在执行数据流图的系统外部的数据源处可用的数据的节点的示例。

[0120] 当执行所生成(640)的数据流图时,可以在所识别的执行系统上执行基于数据流图的程序代码。此外,可以从数据源接收数据。例如,可以如图6C所示地接收(624)数据。

[0121] 在过程600、610、620、630的一些实现中,使用API。在使用API的情况下,过程600包括使得能够响应于接收到SQL查询来执行API的功能。API的功能在外部资源处可执行。此外,(例如,在图5C所示的过程620中)被发送至资源的指令使得能够执行API的功能。以这种方式,SQL查询可以用于查询不是关系数据库管理系统并且不解释SQL查询的资源。

[0122] 图1所示的生成引擎120和/或执行引擎140可例如在合适的操作系统(诸如UNIX操作系统版本等)的控制下托管在一个或多个通用计算机上。例如,这可以包括多节点并行计算环境,其包括使用本地的(例如,多处理器系统,诸如对称多处理(SMP)计算机等)、或本地分布的(例如,耦合为集群或大规模并行处理(MPP)系统的多个处理器)、或远程的、或远程分布的(例如,经由局域网(LAN)和/或广域网(WAN)而耦合的多个处理器)或其任何组合的多个中央处理单元(CPU)或处理器核的计算机系统的配置。

[0123] 图3所示的用户界面300可以是开发环境的一部分。在一些实现中,开发环境是用于开发按照数据流图的应用的系统。根据本系统制作的数据流图提供了用于将信息输入至图组件所表示的各个处理中并从中输出信息、用于使信息在处理之间移动以及用于定义处理的运行顺序的方法。该系统包括从任何可用方法中选择处理间通信方法(例如,根据图链接的通信路径可以使用TCP/IP或UNIX域套接字或者使用共享内存来在处理间传递数据)的算法。

[0124] 上述的技术可以使用执行合适软件的计算系统来实现。例如,该软件可以包括在一个或多个编程或可编程计算系统(可以具有诸如分布式、客户端/服务器或网格等等的各种架构)上执行的一个或多个计算机程序中的过程,其中该一个或多个编程或可编程计算系统各自包括至少一个处理器、至少一个数据存储系统(包括易失性和/或非易失性存储器)和/或存储元件)、至少一个用户接口(用于使用至少一个输入装置或端口来接收输入,并且用于使用至少一个输出装置或端口来提供输出)。该软件可以包括例如提供与数据流图的设计、配置和执行有关的服务的较大程序的一个或多个模块。可以将程序的模块(例如,数据流图的元素)实现为数据结构或符合数据存储库中所存储的数据模型的其它有组织数据。

[0125] 可以将软件设置在诸如(利用通用或专用可编程计算机可读取的)CD-ROM或其它计算机可读介质等的有形、非暂时性介质上、或者经由网络的通信介质(例如,以编码在传播信号中的形式)传递至执行该软件的计算系统的有形、非暂时性介质。可以在专用计算机上、或者使用诸如协处理器或现场可编程门阵列(FPGA)或专用特定用途集成电路(ASIC)等的专用硬件来进行一些处理或所有功能。可以以利用不同的计算元件来进行软件所指定的计算的不同部分的分布式方式来实现该处理。优选将每一个这种计算机程序存储在通用或专用可编程计算机可读取的存储装置的计算机可读存储介质(例如,固态存储器或介质、或者磁性或光学介质)上或者下载至该存储介质,以在利用计算机读取存储装置介质以进行

这里所述的处理的情况下配置计算机并使该计算机进行工作。本发明的系统还可被视为作为配置有计算机程序的有形、非暂时性介质来实现,其中如此配置成的介质使计算机以特定的预定义方式进行工作,以进行这里所述的处理步骤中的一个或多个。

[0126] 已经描述了本发明的许多实施例。然而,应当理解,前述描述旨在示出而不是限制本发明的范围,该范围由所附权利要求书的范围限定。因此,其它实施例也在所附权利要求书的范围内。例如,可以在不偏离本发明的范围的情况下作出各种修改。另外,上述步骤中的一些可以是与顺序无关的,因此可以按照与所描述的顺序不同的顺序执行。

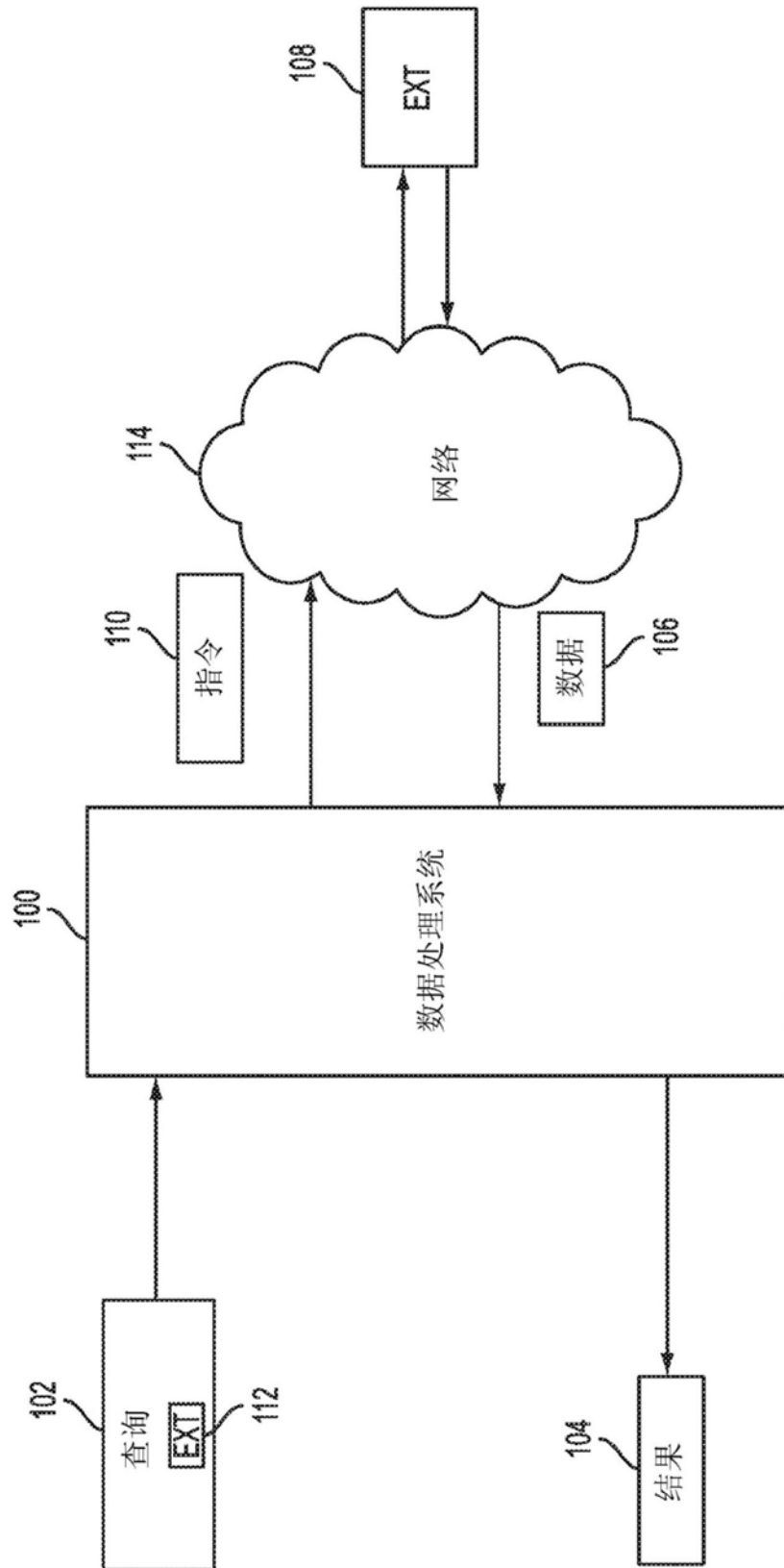


图1

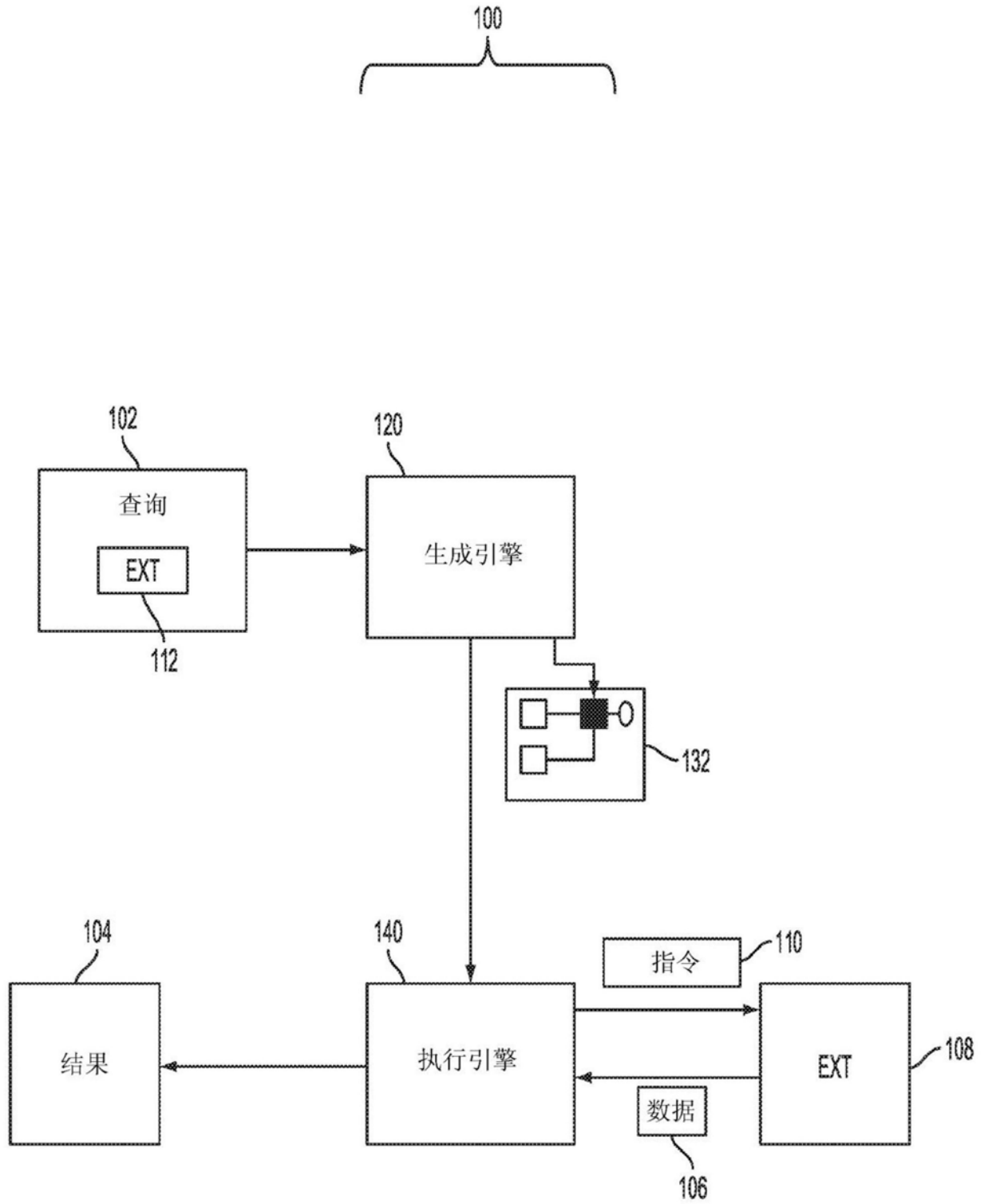


图2A

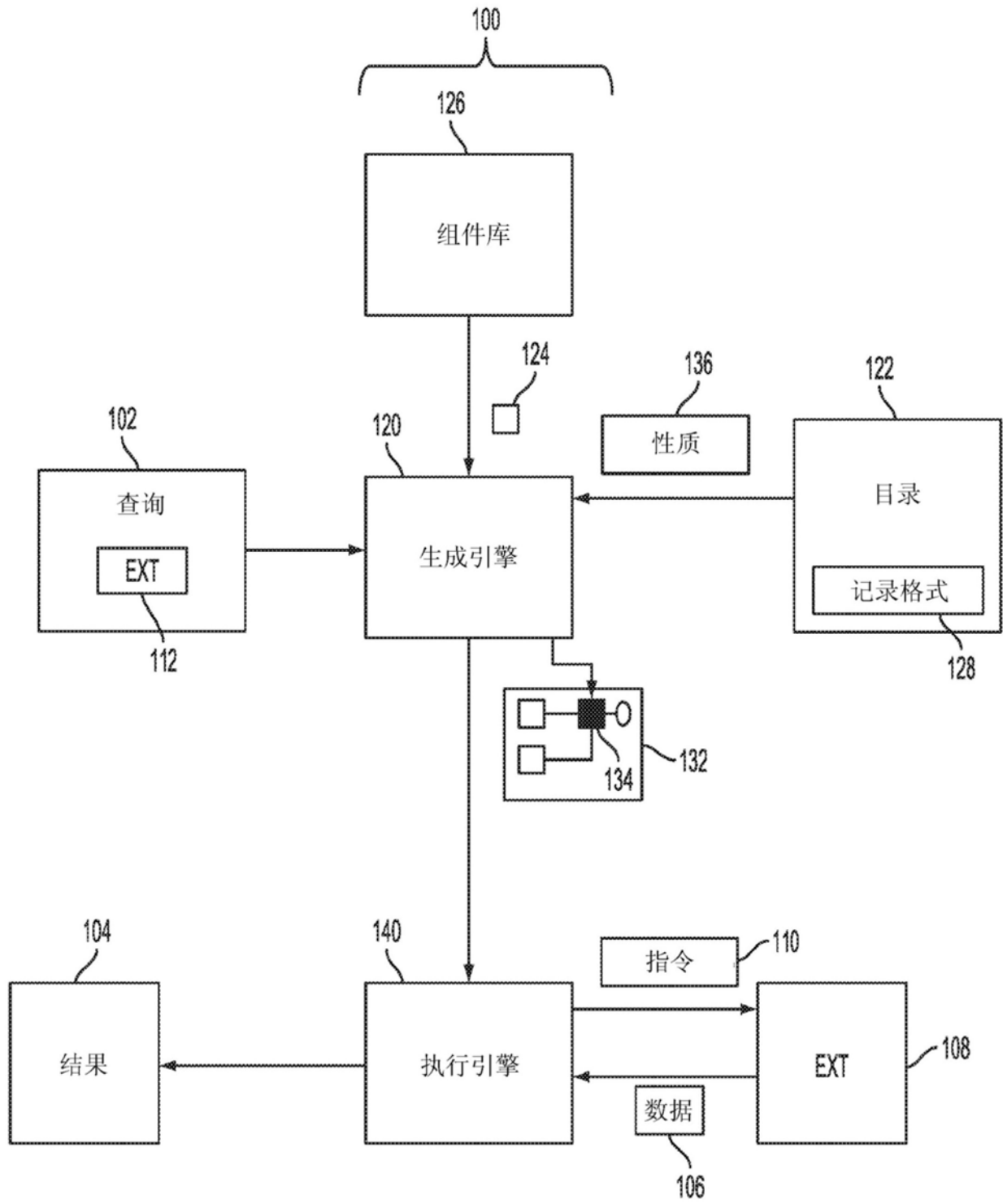


图2B

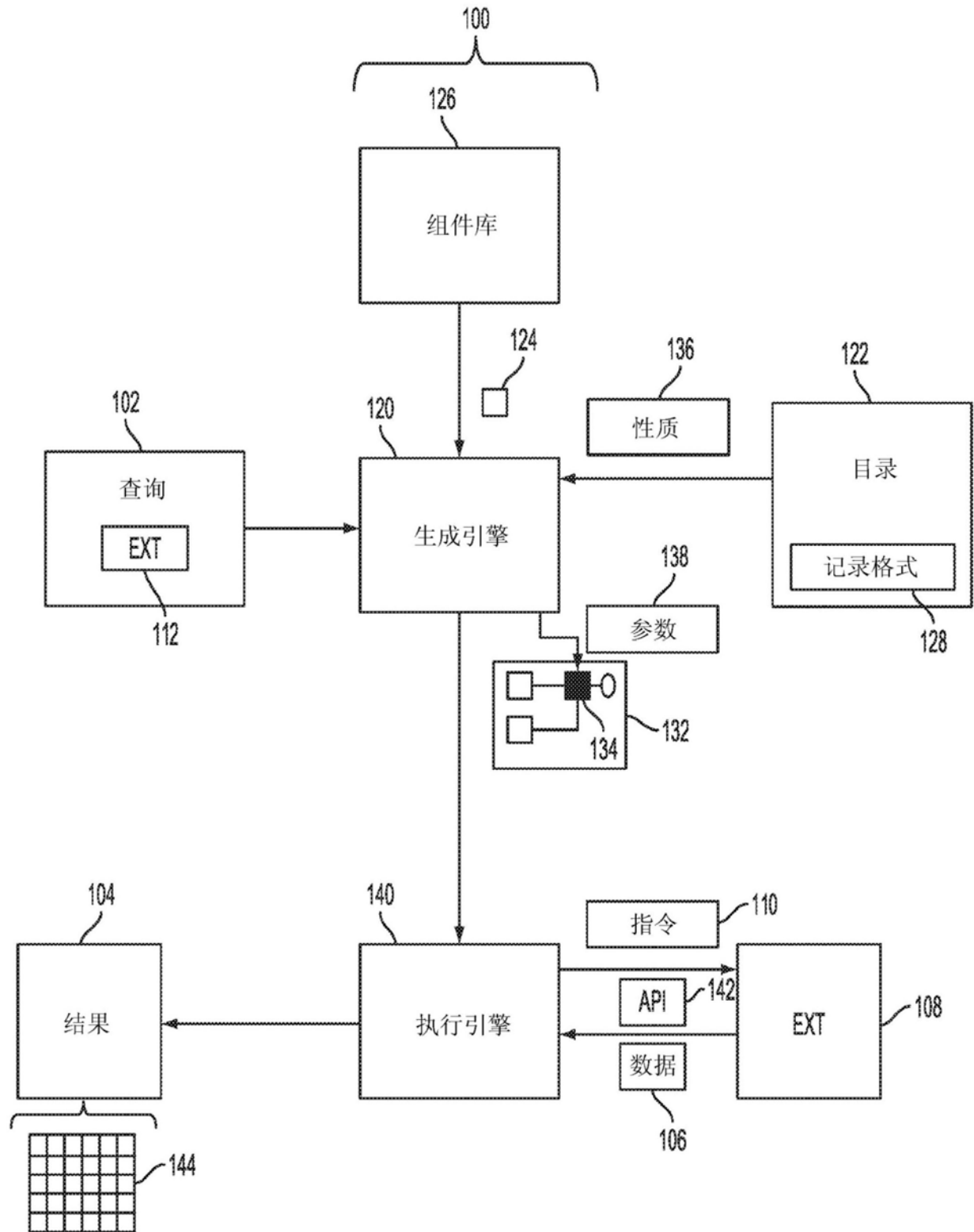


图2C

300

> Ab Initio ODBC 查看器 (v2.0.1)

输入DSN名称: 本地主机

浏览DSN

断开连接

查询: `SELECT * from web.wikipedia where subject = 'SQL'`

跟踪文件

跟踪级别

查看配置

最后一个错误

最大行 500

迭代 1(单个查询)

☐ 显示类型

显示表 显示类型 显示模式 连接信息 运行查询

306 308 318 320 302 304 310 316

主题	行
SQL	SQL (/ "ss kju: 'sl/ "S-Q-L"; [4]) (结构化查询语言[5] [6] [7] [8])是被设计用于管理保存在关系数据库管理系统 (RDBMS) 中的数据的数据的专用编程语言。
SQL	最初基于关系代数和元组关系演算, SQL 包含数据定义和数据操作语言。SQL 的范围包括数据插入、查询、更新和删除、模式创建和修改、以及数据访问控制。尽管SQL通常被描述为并且很大程度上是一种声明性语言(4GL), 但其也包括过程元素。
SQL	SQL 是埃德加•F•科德 (Edgar F. Codd) 的关系模型的第一商业语言之一, 如他的有影响力的 1970 报纸 "A Relational Model of Data for Large Shared Data Banks" 中所述。[9] 尽管不全部坚持科德所述的关系模型, 但他的关系模型变成了最广泛使用的数据库语言。[10] [11]
SQL	SQL 在 1986 年变成美国国家标准学会 (ANSI) 的标准, 并且在 1987 年变成国际标准化组织 (ISO) 之一的标准。[12] 此后, 该标准已经增加了几倍, 并且具有增加的特征。但是代码在不同的数据库系统之中不是完全可移植的, 这样可能会造成厂商锁定。不同的标记不完全遵守标准, 它们会添加扩展, 并且标准有时是不明确的。

图3

314	SQL	SQL 最初是由 IBM 的唐纳德 D 张伯伦 (Donald D. Chamberlin) 和雷蒙德 F 博伊斯 (Raymond F. Boyce) 在 20 世纪 70 年代早期开发出来的。这一版本 (最初称为 SEQUEL (结构化英语查询语言)) 被设计为操作并检索存储在 IBM 最初的准关系数据库管理系统 (System R) 中的数据, 其中, 该准关系数据库管理系统是由 IBM 圣何塞研究实验室的一个小组在 20 世纪 70 年代开发出来的。[13] 首字母缩略词 SEQUEL 后来变为 SQL, 因为 “SEQUEL” 是英国霍克·西德利航空公司的商标。[14]
	SQL	在 20 世纪 70 年代后期, 关系软件有限公司 (现称为甲骨文公司) 见证了科德、张伯伦和博伊斯所述概念的潜势, 并且开发了他们自己的基于 SQL 的 RQBMS, 希望将其卖给美国海军中央情报局以及其它美国政府机构。在 1979 年 6 月, 关系软件有限公司引入了第一个市售的 SQL 实现, 针对 VAX 计算机的 Oracle V2 (第 2 版)。
	SQL	在客户测试网站测试 SQL 以确定系统的可用性和实用性之后, IBM 开始基于包括分别在 1979、1981 和 1983 年市售的 System/38、SQL/DS 以及 DB2 的 System R 原型来开发商业产品。[15]
	SQL	SQL 语言被细分为若干语言元素, 包括:
	SQL	SQL 具有 case/when/then/else/end 表达式, 这些表达式是在 SQL-92 中引入的。在其最普遍的形式 (在 SQL 标准中被称为 “searched case”) 中, searched case 的用法与其它编程语言中的 else if 相同:
	SQL	以 WHEN 条件出现在源中的顺序来测试这些 WHEN 条件。如果未指定 ELSE 表达式, 则默认为 ELSE NULL。存在反映 switch 语句的缩略语法, 在 SQL 标准下称作 “simple case”:
	SQL	这种语法使用隐式相等比较, 并且用于比较的常见提醒为空。
	SQL	对于 Oracle-SQL 方言, 后者可以缩短为等效的 DECODE 结构:
	SQL	最后一个值是默认的; 如果没有进行指定, 则也默认为空。然而, 不同于该标准的 “simple case”, Oracle 的 DECODE 认为两个空值彼此相等。
状态: [56 rows] - 查询: 'SELECT * from web.wikipedia where subject = 'SQL" 完成查询时间: 3.360 mSec 渲染时间: 5mSec		

312

图3 (续)



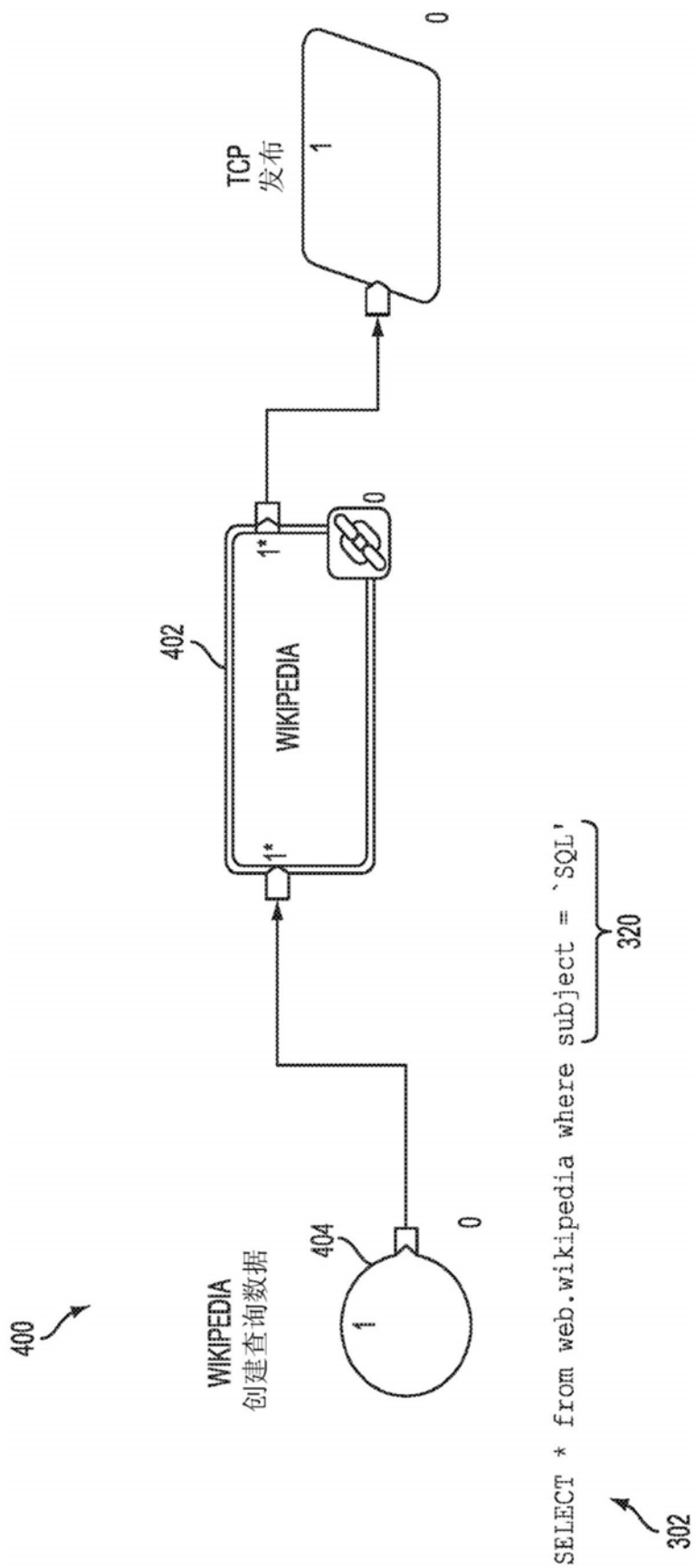


图4A

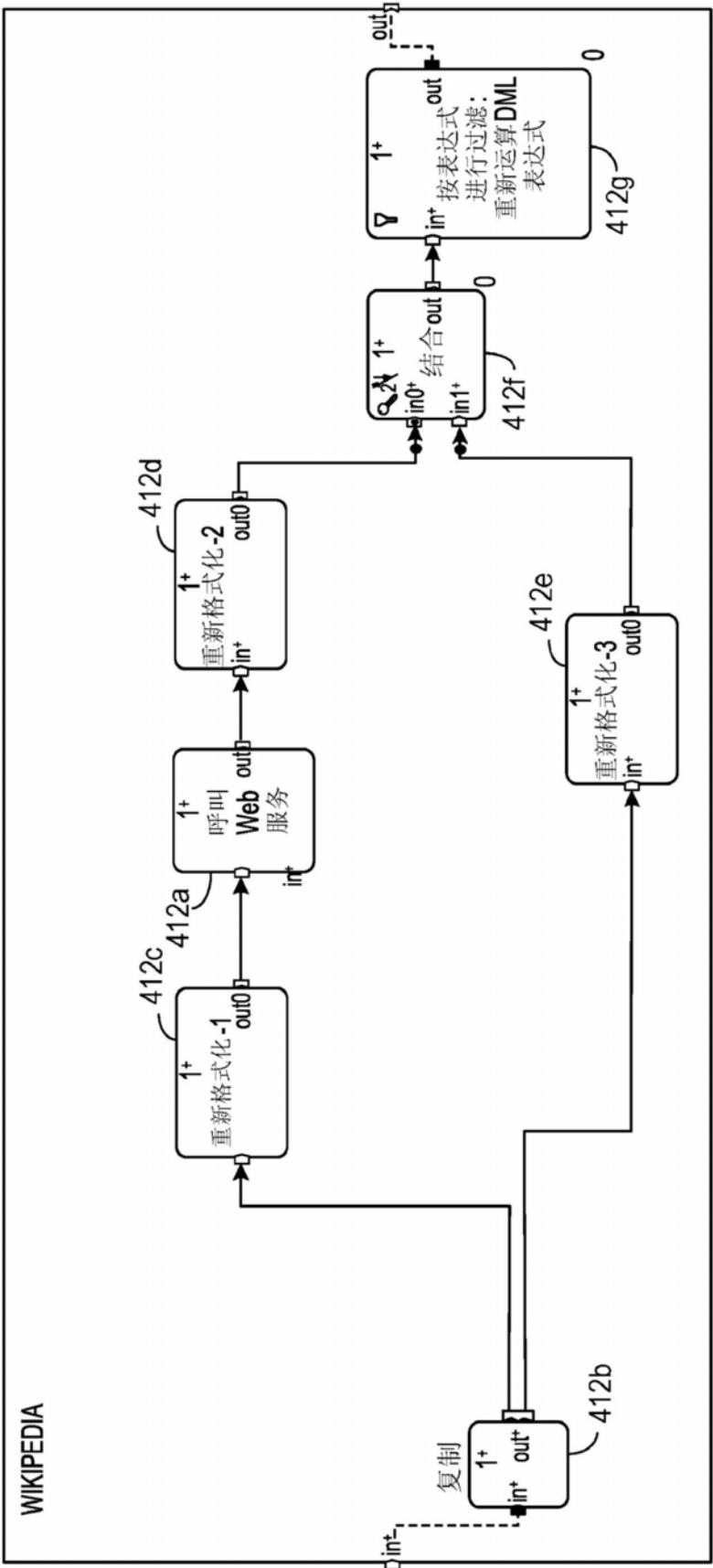


图4B

```

1 type request_type=record
2   ascii string ("\0" request_url;
3   record
4     utf8 string ("\0" name;
5     utf8 string ("\0" value;
6     end [big endian integer (4)] request_url params = NULL;
7     void (big endian integer (4)) request_body = NULL;
8   end; /*Metadata for records containing request information*/
9
10 type response_type=record
11   // The response_type field definitions are shown below.
12   // The definitions are shown here for informational purposes only.
13   // You cannot change the definition of response_type.
14   big endian integer (4) response_code;
15   ascii string (big endian integer (4)) response_headers = NULL;
16   void (big endian integer (4)) response_body = NULL;
17   end; /*Metadata for records containing response information*/
18
19 type response_vector-type=response_type [int]; /*Response vector*/
20
21 request :: prepare_request (in) =
22   begin
23     request.request_url :: "http://en.wikipedia.org/wiki/" + in.subject;
24   end;
25
26 out :: create_output (in, response) =
27   begin
28     out.* :: response.*;
29     out.subject :: in.subject;
30   end;

```

Diagram annotations: A bracket labeled 442 spans lines 3-8. A bracket labeled 444 spans lines 14-17. A bracket labeled 446 spans lines 19-24. A bracket labeled 448 spans lines 27-30.

图4C

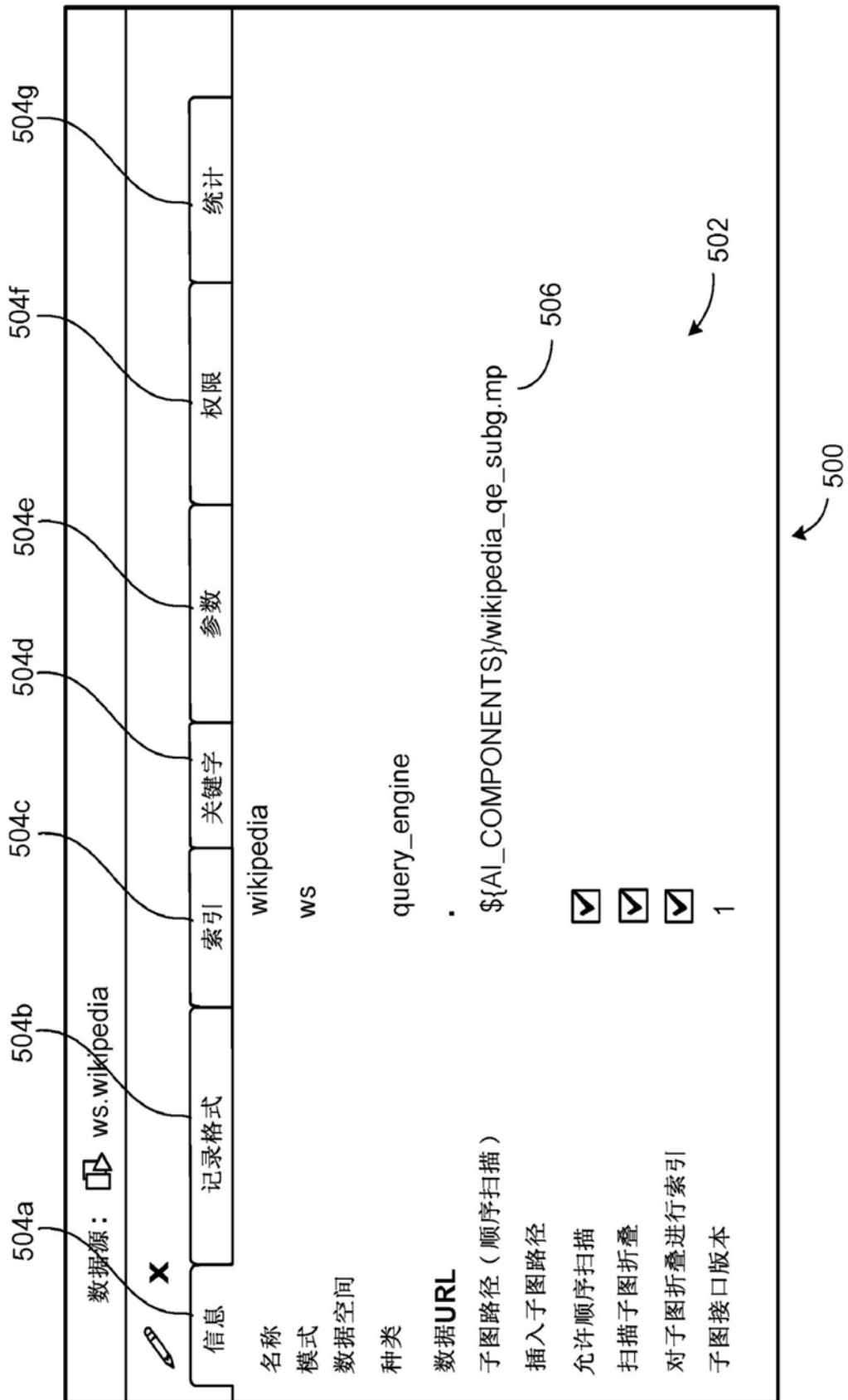




图5A

数据源： ws.wikipedia

 x

记录格式


索引

关键字

参数

权限


统计


 嵌入记录格式


record utf8 string ("") subject; utf8 string ("") line; end

508

) 记录格式文件





 清除

 查看

字段：

510

名称	类型	大小
 主题	字符串	标定为空
 行	字符串	标定为空

500

502

图5B



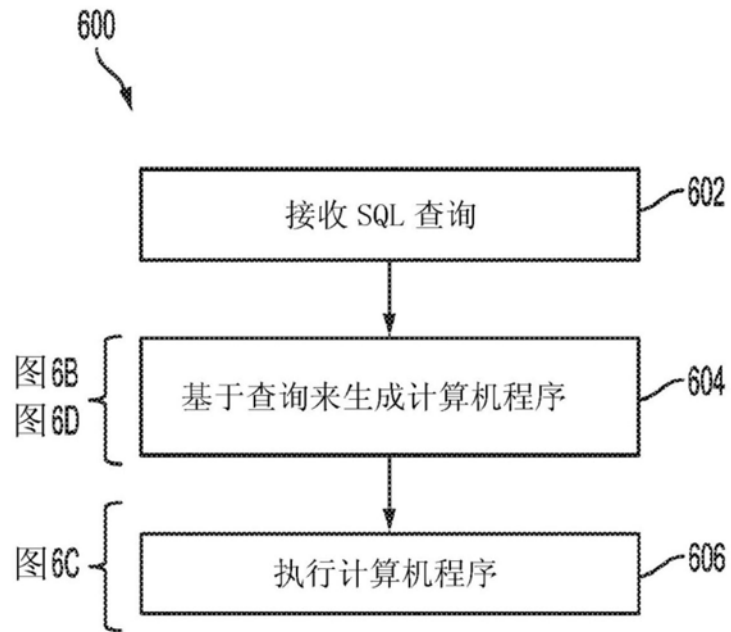


图6A

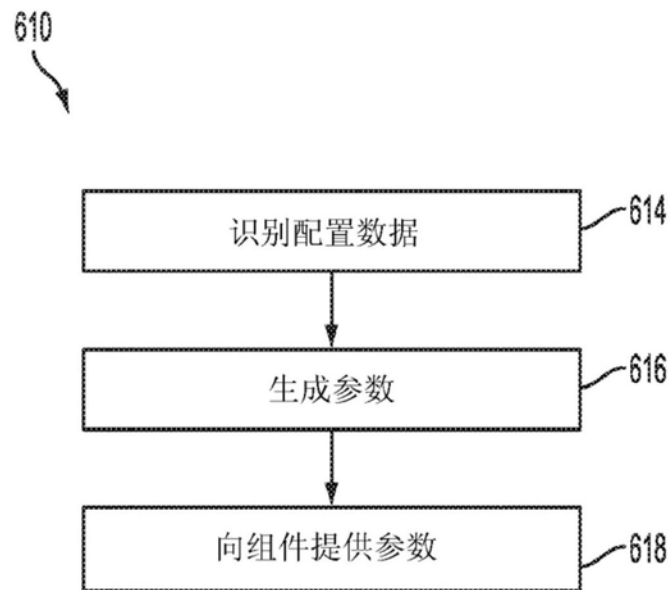


图6B

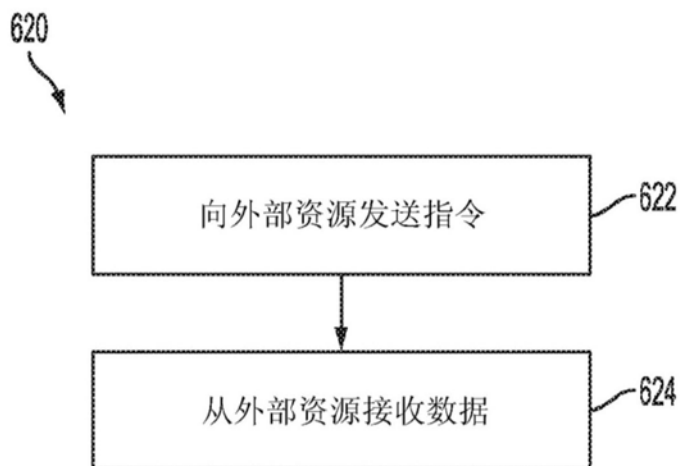


图6C

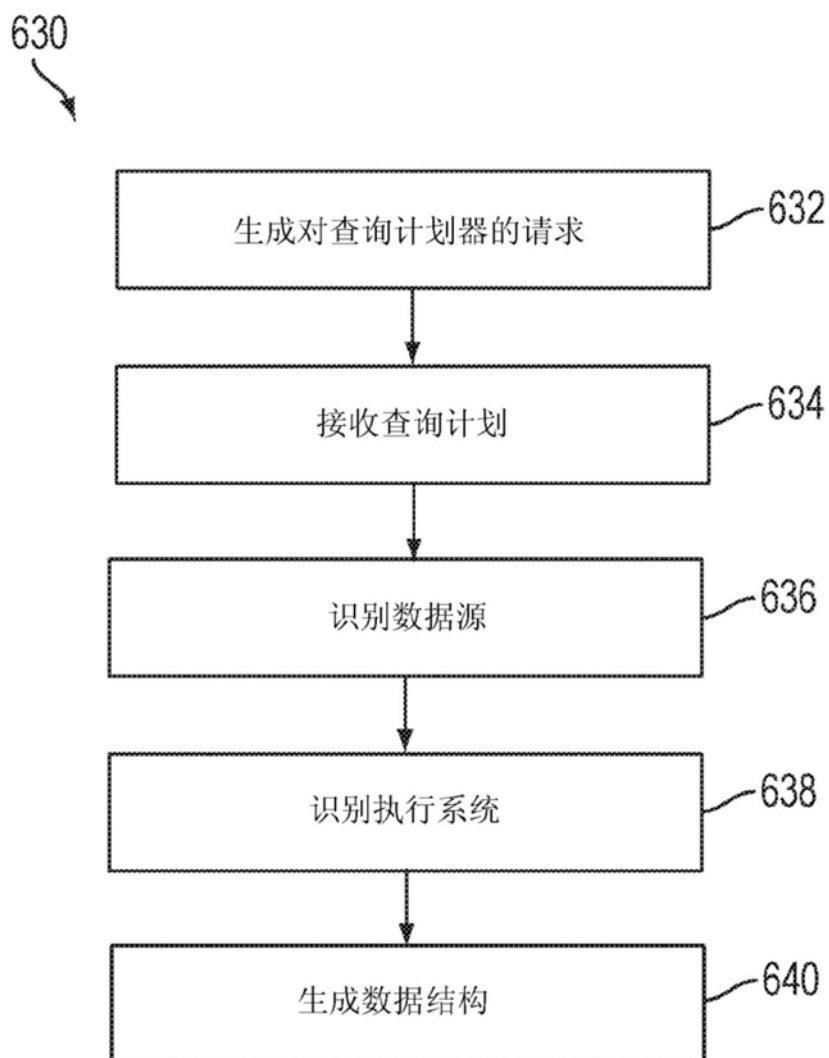


图6D