(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119201972 A
(43)申请公布日 2024.12.27

(21)申请号 202411261534.X

(22)申请日 2024.09.10

(71)申请人 中国建设银行股份有限公司
地址 100033 北京市西城区金融大街25号
申请人 建信金融科技有限责任公司

(72)发明人 游屹

(74)专利代理机构 北京同达信恒知识产权代理有限公司 11291
专利代理师 姚晓雨

(51)Int.Cl.
G06F 16/242(2019.01)
G06F 16/22(2019.01)
G06F 16/28(2019.01)

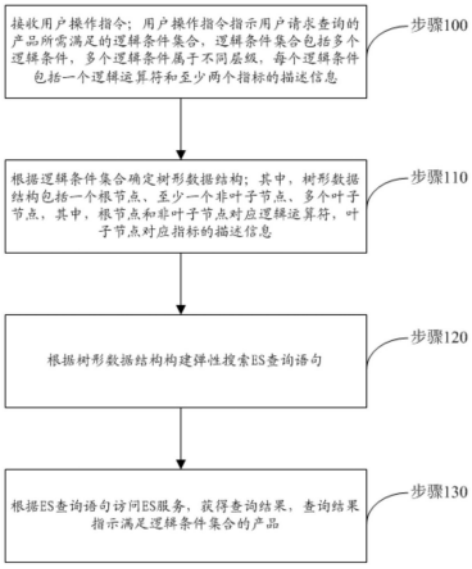
权利要求书2页 说明书17页 附图4页

(54)发明名称

一种搜索方法、装置、设备及存储介质

(57)摘要

本申请涉及数据处理领域,具体涉及一种搜索方法、装置、设备及存储介质,该方法包括:接收用户操作指令;用户操作指令指示用户请求查询的产品所需满足的逻辑条件集合,逻辑条件集合中的多个逻辑条件属于不同层级,每个逻辑条件包括一个逻辑运算符和至少两个指标的描述信息;根据逻辑条件集合确定树形数据结构;树形数据结构包括一个根节点、至少一个非叶子节点、多个叶子节点,根节点和非叶子节点对应逻辑运算符,叶子节点对应指标的描述信息;根据树形数据结构构建ES查询语句,根据ES查询语句访问ES服务,获得查询结果,查询结果指示满足逻辑条件集合的产品。上述方法适用于复杂查询的场景,有效解决了复杂查询精度低的问题。



CN 119201972 A

1.一种搜索方法,其特征在于,该方法包括:

接收用户操作指令;所述用户操作指令指示用户请求查询的产品所需满足的逻辑条件集合,所述逻辑条件集合包括多个逻辑条件,所述多个逻辑条件属于不同层级,每个逻辑条件包括一个逻辑运算符和至少两个指标的描述信息;

根据所述逻辑条件集合确定树形数据结构;其中,所述树形数据结构包括一个根节点、至少一个非叶子节点、多个叶子节点,其中,所述根节点和所述非叶子节点对应逻辑运算符,所述叶子节点对应指标的描述信息;

根据所述树形数据结构构建弹性搜索ES查询语句;

根据所述ES查询语句访问ES服务,获得查询结果,所述查询结果指示满足所述逻辑条件集合的产品。

2.如权利要求1所述的方法,其特征在于,在所述树形数据结构中,所述根节点与至少两个叶子节点具有关联关系、所述根节点与至少一个非叶子节点连接,每个非叶子节点与至少两个叶子节点具有关联关系。

3.如权利要求2所述的方法,其特征在于,根据所述树形数据结构构建ES查询语句,包括:

确定每个叶子节点对应的子句;

根据所述根节点对应的逻辑运算符和与所述根节点关联的至少两个叶子节点分别对应的子句确定第一ES查询子句;

根据每个非叶子节点对应的逻辑运算符和与所述非叶子节点关联的至少两个叶子节点分别对应的子句确定第二ES查询子句;

根据所述第一ES查询子句和所述至少一个非叶子节点分别对应的第二ES查询子句按照所述树形数据结构自顶向下顺序组装获得所述ES查询语句。

4.如权利要求3所述的方法,其特征在于,每个指标的描述信息包括指标的名称、指标的筛选方式和相应的筛选值;

确定每个叶子节点对应的子句,包括:

根据映射关系和每个叶子节点对应的指标的描述信息中的所述指标的名称确定相应的ES字段信息;所述映射关系指示N个指标名称与N个ES字段信息一一对应,N为正整数;

根据每个叶子节点对应的指标的描述信息中的筛选方式确定相应的装配规则;

根据获得的ES字段信息、所述装配规则以及所述筛选值确定该叶子节点对应的子句。

5.如权利要求4所述的方法,其特征在于,所述映射关系根据产品信息集合构建,所述N个指标名称根据所述产品信息集合确定。

6.如权利要求4所述的方法,其特征在于,所述筛选方式包括如下任意一项:等值匹配、多值等值匹配、不等式匹配、模糊匹配、搜索匹配。

7.如权利要求3所述的方法,其特征在于,还包括:

在所述根节点对应的逻辑运算符与第一非叶子节点对应逻辑运算符相同,且所述第一非叶子节点与所述根节点连接时,将所述第一ES查询子句与所述第一非叶子节点对应的第二ES查询子句进行合并处理。

8.一种搜索装置,其特征在于,该装置包括:通信单元和处理单元;

所述通信单元,用于接收用户操作指令;所述用户操作指令指示用户请求查询的产品

所需满足的逻辑条件集合,所述逻辑条件集合包括多个逻辑条件,所述多个逻辑条件属于不同层级,每个逻辑条件包括一个逻辑运算符和至少两个指标的描述信息;

所述处理单元,用于根据所述逻辑条件集合确定树形数据结构;其中,所述树形数据结构包括一个根节点、至少一个非叶子节点、多个叶子节点,其中,所述根节点和所述非叶子节点对应逻辑运算符,所述叶子节点对应指标的描述信息;根据所述树形数据结构构建ES查询语句;根据所述ES查询语句访问ES服务,获得查询结果,所述查询结果指示满足所述逻辑条件集合的产品。

9.一种搜索设备,其特征在于,包括:

存储器,用于存储程序指令;

处理器,用于调用所述存储器中存储的程序指令,按照获得的程序指令执行权利要求1-7中任一项所述的方法包括的步骤。

10.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储有计算机程序,所述计算机程序包括程序指令,所述程序指令当被计算机执行时,使所述计算机执行如权利要求1-7中任一项所述的方法。

11.一种计算机程序产品,其特征在于,所述计算机程序产品包括:计算机程序代码,当所述计算机程序代码在计算机上运行时,使得计算机执行上述如权利要求1-7中任一项所述的方法。

一种搜索方法、装置、设备及存储介质

技术领域

[0001] 本申请涉及数据处理领域,具体涉及一种搜索方法、装置、设备及存储介质。

背景技术

[0002] 随着信息技术的快速发展,特别是在金融行业,数据量的爆炸性增长带来了巨大的挑战和机遇。金融产品的种类和数量日益增多,涉及的标签和指标也日趋复杂。因此,业务人员需要从海量的金融产品中快速准确地筛选出符合特定条件的产品以满足客户需求。目前,金融产品信息的检索主要依赖传统的关系型数据库或简单的搜索引擎,尽管这些技术已经相当成熟,但在处理大规模、多维度、实时性要求高的数据检索任务时仍存在不足。例如,关系型数据库在执行全文检索和复杂逻辑查询时性能较差,而通用的搜索引擎无法满足数据查询高精度的需求。

[0003] 弹性搜索(Elasticsearch,ES)作为一种高性能的搜索引擎,拥有强大的全文检索能力、灵活的数据处理特性以及较高的实时性。当检索条件繁多并且复杂时,如何基于ES的检索方法提升查询精度是一个值得关注的问题。

发明内容

[0004] 本申请实施例提供一种搜索方法、装置、设备及存储介质,用于提升查询精度。

[0005] 第一方面,本申请提供一种搜索方法,所述方法包括:

[0006] 接收用户操作指令;所述用户操作指令指示用户请求查询的产品所需满足的逻辑条件集合,所述逻辑条件集合包括多个逻辑条件,所述多个逻辑条件属于不同层级,每个逻辑条件包括一个逻辑运算符和至少两个指标的描述信息;

[0007] 根据所述逻辑条件集合确定树形数据结构;其中,所述树形数据结构包括一个根节点、至少一个非叶子节点、多个叶子节点,其中,所述根节点和所述非叶子节点对应逻辑运算符,所述叶子节点对应指标的描述信息;

[0008] 根据所述树形数据结构构建ES查询语句;

[0009] 根据所述ES查询语句访问ES服务,获得查询结果,所述查询结果指示满足所述逻辑条件集合的产品。

[0010] 本发明实施例中,提供一种搜索方法,将用户操作指令指示的逻辑条件集合转换成树形数据结构,逻辑条件集合中的多个逻辑条件可以属于不同层级,上述方法可以应用于比较复杂的查询场景,能够同时考虑多个逻辑条件,利用了ES的全文检索和数据分析能力,能够快速查询到满足逻辑条件集合的产品,提高了查询精度。

[0011] 在一种可能的实施例中,在所述树形数据结构中,所述根节点与至少两个叶子节点具有关联关系、所述根节点与至少一个非叶子节点连接,每个非叶子节点与至少两个叶子节点具有关联关系。

[0012] 在一种可能的实施例中,根据所述树形数据结构构建ES查询语句,包括:

[0013] 确定每个叶子节点对应的子句;

[0014] 根据所述根节点对应的逻辑运算符和与所述根节点关联的至少两个叶子节点分别对应的子句确定第一ES查询子句;

[0015] 根据每个非叶子节点对应的逻辑运算符和与所述非叶子节点关联的至少两个叶子节点分别对应的子句确定第二ES查询子句;

[0016] 根据所述第一ES查询子句和所述至少一个非叶子节点分别对应的第二ES查询子句按照所述树形数据结构自顶向下顺序组装获得所述ES查询语句。

[0017] 在本实施例中,通过解析树形数据结构,可以确定每个叶子节点对应的子句,根据根节点对应的逻辑运算符和与根节点关联的至少两个叶子节点分别对应的子句确定第一ES查询子句,以及根据非叶子节点对应的逻辑运算符和与所述非叶子节点关联的至少两个叶子节点分别对应的子句确定第二ES查询子句,将第一查询子句和各个第二查询子句组装起来,不仅能够实现对这个树形数据结构自上向下的全面解析,而且确保了查询语句的结构完整性,减少了误差和遗漏。

[0018] 在一种可能的实施例中,每个指标的描述信息包括指示指标的名称、一个指标的筛选方式和相应的筛选值;

[0019] 确定每个叶子节点对应的子句,包括:

[0020] 根据映射关系和每个叶子节点对应的指标的描述信息中的所述指标的名称确定相应的ES字段信息;所述映射关系指示N个指标名称与N个ES字段信息一一对应,N为正整数;

[0021] 根据每个叶子节点对应的指标的描述信息中的筛选方式和筛选值确定相应的装配规则;

[0022] 根据获得的ES字段信息、所述和装配规则以及所述筛选值确定该叶子节点对应的子句。

[0023] 在本实施例中,通过构建指标名称和ES字段信息的映射关系,用户只需要在可视化界面上灵活选择指标名称,就可以直接关联到ES的字段信息,不仅简化了用户的操作流程,且非技术用户也能够轻松地构建逻辑条件集合,有效降低了技术门槛。

[0024] 在一种可能的实施例中,所述映射关系根据产品信息集合构建,所述N个指标名称根据所述产品信息集合确定。

[0025] 在一种可能的实施例中,所述筛选方式包括如下任意一项:等值匹配、多值等值匹配、不等式匹配、模糊匹配、搜索匹配。

[0026] 在一种可能的实施例中,还包括:

[0027] 在所述根节点对应的逻辑运算符与第一非叶子节点对应逻辑运算符相同,且所述第一非叶子节点与所述根节点连接时,将所述第一ES查询子句与所述第一非叶子节点对应的第二ES查询子句进行合并处理。

[0028] 在本实施例中,当服务器接收到传送的复杂查询请求时,通过上述合并查询方式,可以减少查询的深度,简化查询结构,从而提高查询效率。

[0029] 第二方面,本申请还提供一种搜索装置,所述装置包括:接收单元和处理单元;

[0030] 所述接收单元,用于接收用户操作指令;所述用户操作指令指示用户请求查询的产品所需满足的逻辑条件集合,所述逻辑条件集合包括多个逻辑条件,所述多个逻辑条件属于不同层级,每个逻辑条件包括一个逻辑运算符和至少两个指标的描述信息;

[0031] 所述处理单元,用于根据所述逻辑条件集合确定树形数据结构;其中,所述树形数据结构包括一个根节点、至少一个非叶子节点、多个叶子节点,其中,所述根节点和所述非叶子节点对应逻辑运算符,所述叶子节点对应指标的描述信息;根据所述树形数据结构构建弹性搜索ES查询语句;根据所述ES查询语句访问ES服务,获得查询结果,所述查询结果指示满足所述逻辑条件集合的产品。

[0032] 在一种可能的实施例中,所述树形数据结构中,所述根节点与至少两个叶子节点具有关联关系、所述根节点与至少一个非叶子节点连接,每个非叶子节点与至少两个叶子节点具有关联关系;

[0033] 在一种可能的实施例中,所述处理单元在根据所述树形数据结构构建弹性搜索ES查询语句时,具体用于:

[0034] 确定每个叶子节点对应的子句;

[0035] 根据所述根节点对应的逻辑运算符和与所述根节点关联的至少两个叶子节点分别对应的子句确定第一ES查询子句;

[0036] 根据每个非叶子节点对应的逻辑运算符和与所述非叶子节点关联的至少两个叶子节点分别对应的子句确定第二ES查询子句;

[0037] 根据所述第一ES查询子句和所述至少一个非叶子节点分别对应的第二ES查询子句按照所述树形数据结构自顶向下顺序组装获得所述ES查询语句。

[0038] 在一种可能的实施例中,每个指标的描述信息包括指标的名称、指标的筛选方式和相应的筛选值;所述处理单元在确定每个叶子节点对应的子句时,具体用于:

[0039] 根据映射关系和每个叶子节点对应的指标的描述信息中的所述指标的名称确定相应的ES字段信息;所述映射关系指示N个指标名称与N个ES字段信息一一对应,N为正整数;

[0040] 根据每个叶子节点对应的指标的描述信息中的筛选方式确定相应的装配规则;

[0041] 根据获得的ES字段信息、所述装配规则以及所述筛选值确定该叶子节点对应的子句。

[0042] 在一种可能的实施例中,所述映射关系根据产品信息集合构建,所述N个指标名称根据所述产品信息集合确定。

[0043] 在一种可能的实施例中,所述筛选方式包括如下任意一项:等值匹配、多值等值匹配、不等式匹配、模糊匹配、搜索匹配。

[0044] 在一种可能的实施例中,所述处理单元,还包括:

[0045] 在所述根节点对应的逻辑运算符与第一非叶子节点对应逻辑运算符相同,且所述第一非叶子节点与所述根节点连接时,将所述第一ES查询子句与所述第一非叶子节点对应的第二ES查询子句进行合并处理。

[0046] 第三方面,本申请提供一种搜索设备,包括:

[0047] 存储器,用于存储程序指令;

[0048] 处理器,用于调用所述存储器中存储的程序指令,按照获得的程序指令执行第一方面中任一项所述的方法包括的步骤。

[0049] 第四方面,本申请提供一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序包括程序指令,所述程序指令当被计算机执行时,使所述计

算机执行第一方面中任一项所述的方法。

[0050] 第五方面,本申请提供了一种计算机程序产品,所述计算机程序产品包括:计算机程序代码,当所述计算机程序代码在计算机上运行时,使得计算机执行第一方面中任一项所述的方法。

附图说明

[0051] 图1为本申请实施例提供的一种搜索方法的流程图;

[0052] 图2为本申请实施例提供的一种可视化界面的示意图;

[0053] 图3A为本申请实施例提供的一种可能的树形数据结构的示意图一;

[0054] 图3B为本申请实施例提供的一种可能的树形数据结构的示意图二;

[0055] 图4为本申请实施例提供的搜索装置的结构示意图一;

[0056] 图5为本申请实施例提供的搜索装置的结构示意图二。

具体实施方式

[0057] 为使本申请的目的、技术方案和优点更加清楚明白,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。在不冲突的情况下,本申请中的实施例及实施例中的特征可以相互任意组合。并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以按不同于此处的顺序执行所示出或描述的步骤。

[0058] 本申请的说明书和权利要求书及上述附图中的术语“第一”和“第二”是用于区别不同对象,而非用于描述特定顺序。此外,术语“包括”以及它们任何变形,意图在于覆盖不排他的保护。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备没有限定于已列出的步骤或单元,而是可选地还包括没有列出的步骤或单元,或可选地还包括对于这些过程、方法、产品或设备固有的其它步骤或单元。本申请中的“多个”可以表示至少两个,例如可以是两个、三个或者更多个,本申请实施例不做限制。

[0059] 以下结合附图对本申请的示范性实施例做出说明,其中包括本申请实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本申请公开的范围。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。需要说明的是,在本申请实施例中,可能提及某些软件、组件、模型等业界已有方案,应当将它们认为是示范性的,其目的仅仅是为了说明本申请技术方案实施中的可行性,但并不意味着申请人已经或者必然用到了该方案。

[0060] 本申请技术方案中对数据的获取、传输、存储、使用等,均符合国家相关法律法规要求。

[0061] 需要说明的是,在本申请实施例中,可能提及某些软件、组件、模型等业界已有方案,应当将它们认为是示范性的,其目的仅仅是为了说明本申请技术方案实施中的可行性,但并不意味着申请人已经或者必然用到了该方案。

[0062] 为了便于理解,首先对下面对本申请实施例的技术背景进行详细介绍。

[0063] 1、ES,又可称为弹性搜索,或者搜索数据分析引擎,或者搜索服务器等。ES是一个分布式全文检索引擎,可以提供实时的存储、数据检索和分析能力,能可以扩展部署到上百台服务器,处理万级别的数据。他提供了RESTful接口供使用者调用,其中,RESTful一般指rest,REST即表述性状态传递(Representational State Transfer,REST)。

[0064] ES查询语句:ES的查询语句为JSON语句,语句中定义要查询的索引,查询的方式和查询条件,其中,JSON(JavaScript Object Notation)是一种轻量级的数据交换格式,以文本形式表示结构化的数据。

[0065] 现有技术中,基于ES的检索方法要求使用人员具备结构化查询语句基础,并对ES索引结构、字段类型、数据的组织方式都需要了解,这限制了业务人员和非技术用户的使用,增加了技术门槛。此外,当检索条件繁多并且复杂时,基于ES的检索方法也存在查询精度低、响应时间长的问题。

[0066] 基于上述问题,图1示例性的示出了本发明实施例提供的一种搜索方法的流程图,该流程可由服务器执行为例进行说明。

[0067] 如图1所示,该流程具体包括:

[0068] 步骤100:服务器接收用户操作指令;用户操作指令指示用户请求查询的产品所需满足的逻辑条件集合。逻辑条件集合包括多个逻辑条件,并属于不同层级,每个逻辑条件包括一个逻辑运算符和至少两个指标的描述信息。

[0069] 其中,逻辑运算符可以为AND、OR、NOT,具体含义分别为与、或、非。

[0070] 指标的描述信息包括指标的名称、指标的筛选方式和相应的筛选值。

[0071] 指标的名称,可以为多个,例如产品编号、产品名称、标签A、标签B等。

[0072] 其中,筛选方式可以为如下任意一项:

[0073] a)等值匹配

[0074] 其中,等值匹配可以记为“=”,其中,等值匹配可以支持字符、数值、日期等类型;例如,指标的描述信息可以为:标签A= ‘5’ 。

[0075] b)多值等值匹配

[0076] 其中,多值等值匹配可以记为“IN”,“IN”中至少包含两个字符,表示可以满足多个值中的一个;例如,指标的描述信息可以为:标签B IN(‘2’,‘3’),也即标签B可以等于2,也可以等于3。

[0077] c)不等式匹配

[0078] 其中,不等式可以记为“>、≥、<、≤、BETWEEN...AND...”,其中,不等式匹配可以支持数值和日期型的不等式筛选。例如,指标的描述信息可以为:标签C≥ ‘10’ 。

[0079] d)模糊匹配

[0080] 其中,模糊匹配可以记为“LIKE”,其中,通过输入通配符 ‘*’ 定义需要模糊匹配的部分。例如,指标的描述信息可以为:标签G LIKE ‘*货币*’ ,即搜索与 ‘货币’ 语义相关的标签G。

[0081] e)搜索匹配

[0082] 其中,搜索可以记为“SEARCH”,通过输入任意值进行分词搜索。例如,指标的描述信息可以为:产品名称SEARCH ‘零钱’ ,即将零钱进行分词处理,并搜索分词后结果的产品名

称。

[0083] 示例性地,用户可以通过可视化界面选择各个逻辑运算符、指标的名称、指标的筛选方式以及相应的筛选值,进而生成用户请求查询的产品所需满足的逻辑条件集合。

[0084] 示例性地,可视化界面可以分为左右两个部分,其中,左边部分呈现所有可选的指标名称,右边部分呈现用户请求查询的产品所需满足的逻辑条件集合。

[0085] 在一个实施例中,如图2所示,假如每个金融产品都有十个指标,分别对应产品编号、产品名称、产品描述、标签A、标签B、标签C、标签D、标签E、标签F、标签G。用户首先在可视化界面上可以查询到产品的这些指标名称,了解有哪些指标可供选择。

[0086] 假设用户想要查询标签A等于‘5’,标签B等于‘2’或‘3’,以及标签G包含‘货币’两字的产品,那么用户可以依次点击或拖拽这三个指标到指定的筛选框中,同时以下拉框的方式选择所需的筛选方式和筛选值,具体地,将标签A对应的筛选方式选为“=”,对应的筛选值选为‘5’;将标签B对应的筛选方式选为“IN”,对应的筛选值选为‘2’和‘3’;将“标签G”对应的筛选方式选为“LIKE”,对应的筛选值可以通过输入或选择的方式选为‘货币’。最后用逻辑运算符“AND”将这三个指标的描述信息相连,表示筛选同时满足三个指标的描述信息的产品。

[0087] 步骤110:服务器根据逻辑条件集合确定树形数据结构。

[0088] 其中,树形数据结构包括一个根节点、至少一个非叶子节点、多个叶子节点,其中,根节点和非叶子节点对应逻辑运算符,叶子节点对应指标的描述信息。

[0089] 其中,如果用户选择多个逻辑条件,且多个逻辑条件属于不同层级,那么逻辑条件集合会包含多层嵌套,形成树形结构。在树形结构中,根节点代表顶层运算符,可以将逻辑条件集合中的全部逻辑条件包含在一个树形结构中。

[0090] 在一种可能的实现方式中,根节点与至少两个叶子结点具有关联关系,同时与至少一个非叶子节点连接,每个非叶子节点与至少两个叶子结点具有关联关系。例如,如图3A所示,根节点对应的逻辑运算符是“AND”,与根节点连接的两个叶子节点对应的指标的描述信息是“标签A= ‘value’”和“标签B=(‘T1’,‘T2’)”,与根节点连接的两个非叶子节点对应的逻辑运算符分别是“OR”和“AND”,其中,与非叶子节点对应的逻辑运算符“OR”连接的叶子节点对应的指标的描述信息是“标签C≥10”和“标签D=3”,与非叶子节点对应的逻辑运算符“AND”连接的叶子节点对应的指标的描述信息是“产品名称SEARCH‘零钱’”和“标签G LIKE‘*货币*’”。也就是说,根据用户选择的不同逻辑运算符和指标的描述信息,可以形成两个层级的树形数据结构。

[0091] 在一种可能的实现方式中,根节点与至少两个叶子结点具有关联关系,同时与至少一个非叶子节点连接,每个非叶子节点与至少两个叶子结点具有关联关系,且存在非叶子节点连接至少一个非叶子节点。例如,图3B所示,除了图3A上的所示结构,可以在第二层级的非叶子节点(对应的逻辑运算符为“OR”)下面又连接一个非叶子节点(对应的逻辑运算符为“NOT”),处于第三层级,其中,“NOT”连接的叶子节点对应的指标的描述信息分别是“标签E= ‘20150101’”和“标签F≤‘20231231’”。也就是说,这个树形数据结构可以是三个层级,第三层级的逻辑条件嵌套在第二层级的逻辑条件之中。

[0092] 因此,根节点以及连接的至少两个叶子节点构成第一层级;在此基础上,与根节点连接的非叶子节点以及与这些非叶子节点连接的叶子节点构成第二层级;如果在第二个层

级下面存在其他的非叶子节点以及连接的叶子节点,则构成第三层级,以此类推,可以形成多个层级的树形结构。

[0093] 通过上述树形数据结构,将各个逻辑条件分布在多个层级上,可以适用于同时考虑多个逻辑条件的复杂查询场景,提高查询精度。

[0094] 步骤120:服务器根据树形数据结构构建弹性搜索ES查询语句。

[0095] 在构建ES查询语句之前,先构建映射关系;映射关系指示N个指标名称与N个ES字段信息一一对应,N为正整数。

[0096] 例如,如表1所示,映射关系可以指示指标名称与ES字段信息的对应关系;其中,左边这一列可以显示在步骤110中的用户的可视化界面上,表示可供选择的指标名称有哪些,本申请不限定指标名称的数量,N个指标名称根据产品信息集合确定;右边这一列用于表示指标名称映射的对应ES字段信息。

[0097]

指标名称	ES字段信息
产品编号	PD_ID
产品名称	PD_NAME
产品描述	PD_DESC
标签A	LABEL_A
标签B	LABEL_B
标签C	LABEL_C
标签D	LABEL_D
标签E	LABEL_E
标签F	LABEL_F
标签G	LABEL_G

[0098] 表1

[0099] 进一步地,为了丰富这个映射关系,还可以在映射关系中加入其他信息,例如,指标编号、指标类型和ES索引信息等,如表2所示,通过加入这些列的信息,可以让映射关系更丰富、更完善。

[0100]

指标编号	指标名称	指标类型	ES 索引信息	ES 字段信息
P00001	产品编号	字符型	PD_LABEL_INDEX	PD_ID
P00002	产品名称	文本型	PD_LABEL_INDEX	PD_NAME
P00003	产品描述	文本型	PD_LABEL_INDEX	PD_DESC
P00004	标签 A	字符型	PD_LABEL_INDEX	LABEL_A
P00005	标签 B	字符型	PD_LABEL_INDEX	LABEL_B

[0101]

P00006	标签 C	数值型	PD_LABEL_INDEX	LABEL_C
P00007	标签 D	数值型	PD_LABEL_INDEX	LABEL_D
P00008	标签 E	日期型	PD_LABEL_INDEX	LABEL_E
P00009	标签 F	日期型	PD_LABEL_INDEX	LABEL_F
P00010	标签 G	文本型	PD_LABEL_INDEX	LABEL_G

[0102] 表2

[0103] 进一步地,还可以在映射关系中加入ES字段描述和ES字段类型,如表3所示,通过加入这些列的信息,可以使得ES字段信息的内容更明确、更完善。

[0104]

指标编号	指标名称	指标类型	ES 索引信息	ES 字段信息	ES 字段描述	ES 字段类型
P00001	产品编号	字符型	PD_LABEL_INDEX	PD_ID	产品编号(字符型)	doc_id
P00002	产品名称	文本型	PD_LABEL_INDEX	PD_NAME	产品名称(文本型)	text
P00003	产品描述	文本型	PD_LABEL_INDEX	PD_DESC	产品描述(文本型)	text
P00004	标签 A	字符型	PD_LABEL_INDEX	LABEL_A	标签 A(枚举值)	keyword
P00005	标签 B	字符型	PD_LABEL_INDEX	LABEL_B	标签 B(枚举值)	keyword
P00006	标签 C	数值型	PD_LABEL_INDEX	LABEL_C	标签 C(数值型)	keyword
P00007	标签 D	数值型	PD_LABEL_INDEX	LABEL_D	标签 D(数值型)	keyword
P00008	标签 E	日期型	PD_LABEL_INDEX	LABEL_E	标签 E(日期型)	keyword
P00009	标签 F	日期型	PD_LABEL_INDEX	LABEL_F	标签 F(日期型)	keyword
P00010	标签 G	文本型	PD_LABEL_INDEX	LABEL_G	标签 G(文本型)	text

[0105] 表3

[0106] 此外,表3也可以分解为两个表,分别为表2和表4,表2指示映射关系,表4指示ES索引信息的相关内容,包含四列:索引名、ES字段信息、ES字段描述和ES字段类型,用户所选的指标名称可以通过映射到表4查询到更多相关信息。

[0107] 其中,索引名指示ES字段信息的索引名,ES字段信息指示ES字段名称,ES字段描述指示ES字段类型,ES字段类型中的text和keyword分别指示文本类型和关键词类型,如PD_NAME对应文本类型(text),LABEL_A对应关键词类型(Keyword)。其中,text类型用于全文搜索,适用于需要进行全文检索的字段,当执行搜索查询时,text类型字段的内容会被分词器处理,分词器会将文本分割成单独的词或短语,以便进行搜索。Keyword类型用于精确值匹配,适用于需要进行精确匹配的字段,不会被分割成单独的词。

[0108]

索引名	ES字段信息	ES字段描述	ES字段类型

PD_LABEL_INDEX	PD_ID	产品编号(字符型)	doc_id
PD_LABEL_INDEX	PD_NAME	产品名称(文本型)	text
PD_LABEL_INDEX	PD_DESC	产品描述(文本型)	text
PD_LABEL_INDEX	LABEL_A	标签A(枚举值)	keyword
PD_LABEL_INDEX	LABEL_B	标签B(枚举值)	keyword
PD_LABEL_INDEX	LABEL_C	标签C(数值型)	keyword
PD_LABEL_INDEX	LABEL_D	标签D(数值型)	keyword
PD_LABEL_INDEX	LABEL_E	标签E(日期型)	keyword
PD_LABEL_INDEX	LABEL_F	标签F(日期型)	keyword
PD_LABEL_INDEX	LABEL_G	标签G(文本型)	text
PD_LABEL_INDEX	.....	.....	

[0109] 表4

[0110] 示例性地,结合上述映射关系,服务器根据树形数据结构构建ES查询语句的具体过程为:

[0111] 首先,确定每个叶子节点对应的子句。

[0112] (1)根据映射关系和每个叶子节点对应的指标的描述信息中的指标的名称确定相应的ES字段信息。

[0113] 例如,通过查询表1和表2,可以获得指标名称对应的相应ES字段信息。例如,叶子节点对应的指标的描述信息为:标签A= '5',则通过查询表1,知道"标签A"对应的ES字段信息为"LABEL_A"。

[0114] (2)根据每个叶子节点对应的指标的描述信息中的筛选方式确定相应的装配规则。其中,根据筛选方式的不同,装配规则具体如下:

[0115] a)当筛选方式为等值匹配,该筛选方式对应的装配规则为:

[0116] {"ES字段信息":{:"value"}}。

[0117] b)当筛选方式为多值等值匹配,该筛选方式对应的装配规则为:

[0118] {"terms":{"ES字段信息":["value1","value2","....."]}}。

[0119] c)当筛选方式为不等式匹配,该筛选方式对应的装配规则为:

[0120] {"range":{"ES字段信息":{"gt/gte":"value1","lt/lte":"value2"}}。其中">"对应"gt","≥"对应"gte","<"对应"lt","≤"对应"lte"。

[0121] d)当筛选方式为模糊匹配,该筛选方式对应的装配规则为:

[0122] {"wildcard":{"ES字段信息":{"value":"*query_text*"}}。

[0123] e)当筛选方式为搜索匹配,该筛选方式对应的装配规则为:

[0124] {"match":{"ES字段信息":"搜索词"}}。

[0125] (3)根据获得的ES字段信息、装配规则以及筛选值确定该叶子节点对应的子句。具体地,通过将叶子节点对应的指标的描述信息中的指标名称映射的ES索引信息、筛选方式的装配规则以及用户所选的筛选值组合在一起,获得叶子节点对应的子句。

[0126] 示例性的,如图3B所示,叶子节点分别为:

[0127] 标签A= 'value'

[0128] 标签B IN('T1','T2')

[0129] 标签C≥10

[0130] 标签D＝3

[0131] 标签E＝‘20240101’

[0132] 标签F≤‘20241231’

[0133] 产品名称SERACH‘零钱’

[0134] 标签G LIKE‘*货币*’

[0135] 则获得的叶子节点对应的子句分别为：

[0136] {"LABEL_A":{:"5"}}

[0137] {"LABEL_B":["T1","T2"]}

[0138] {"LABEL_C":{"gte":10}}

[0139] {"LABEL_D":3}

[0140] {"LABEL_E":"20150101"}

[0141] {"LABEL_F":{"lte":"20231231"};

[0142] 其次,根据根节点对应的逻辑运算符和与根节点关联的至少两个叶子节点分别对应的子句确定第一ES查询子句。

[0143] 示例性地,根据根节点对应的逻辑运算符确定根节点对应的逻辑运算符对应装配规则,然后根据根节点对应的逻辑运算符对应的装配规则和与根节点关联的至少两个叶子节点分别对应的子句确定第一ES查询子句。

[0144] 例如,可以用bool查询语句进行组装,例如,当逻辑运算符为“AND”,将逻辑运算符“AND”以及与逻辑运算符“AND”关联的至少两个叶子节点对应的子句装配为must子句,具体的装配规则为：

[0145] AND->must子句,

```
{ "bool": {
        "must": [
            与 AND 连接的叶子节点 1 对应的子句,
            与 AND 连接的叶子节点 2 对应的子句,
            ……],
}}
```

[0146]

[0147] 例如,当逻辑运算符为“OR”,将逻辑运算符“OR”以及与逻辑运算符“OR”关联的至少两个叶子节点对应的子句装配为should子句,具体的装配规则为：

[0148] OR->should子句,并设定参数minimum_should_match为1：

[0149] { "bool": {

```
"should": [
    与 OR 连接的叶子节点 1 对应的子句,
    与 OR 连接的叶子节点 2 对应的子句,
    .....],
"minimum_should_match": 1
}}
```

[0150]

[0151] 例如,当逻辑运算符为“NOT”,将逻辑运算符“NOT”以及与逻辑运算符“NOT”关联的至少两个叶子节点对应的子句装配为must_not子句,具体的装配规则为:

[0152] NOT->must_not子句,

```
{ "bool": {
    "must_not": [
        与 NOT 连接的叶子节点 1 对应的子句,
        与 NOT 连接的叶子节点 2 对应的子句,
        .....],
}}
```

[0153]

[0154] 进一步地,结合图3B,根节点对应的逻辑运算符为“AND”,根据根节点对应的逻辑运算符对应的装配规则和与根节点关联的至少两个叶子节点分别对应的子句获得的第一ES查询子句为:

```
{"bool": {
    "must": [
        { "term": {"LABEL_A": "value"} },
        { "terms": {"LABEL_B": ["T1","T2"]},
}}
```

[0155]

[0156] 第三,根据每个非叶子节点对应的逻辑运算符和与非叶子节点关联的至少两个叶子节点分别对应的子句确定第二ES查询子句;

[0157] 示例性地,根据非叶子节点对应的逻辑运算符确定非叶子节点对应的逻辑运算符对应装配规则,然后根据非叶子节点对应的逻辑运算符对应装配规则和与非叶子节点关联的至少两个叶子节点分别对应的子句确定第二ES查询子句。具体的装配规则可以参考上述规则。此外,非叶子节点也可以连接其他非叶子节点,根据其他非叶子节点对应的逻辑运算符对应装配规则和与其他非叶子节点关联的至少两个叶子节点分别对应的子句确定其他第二ES查询子句,以此类推,可以确定多个第二ES查询子句。

[0158] 进一步地,结合图3B,非叶子节点对应的逻辑运算符分别为“OR”、“AND”和“NOT”,

[0159] 当非叶子节点对应的逻辑运算符为“OR”时,根据逻辑运算符“OR”对应装配规则和与非叶子节点关联的至少两个叶子节点分别对应的子句获得的第二ES查询子句为:

[0160] {"bool":{

[0161] "should":[

[0162] {"range":{"LABEL_C":{"gte":10}}},

[0163] {"term":{"LABEL_D":3}},

[0164] }

[0165] 当非叶子节点对应的逻辑运算符为“AND”时,根据逻辑运算符“AND”对应装配规则和与非叶子节点关联的至少两个叶子节点分别对应的子句获得的第二ES查询子句为:

[0166] {"bool":{

[0167] "must":[

[0168] {"match":{"PD_NAME":"零钱"}},

[0169] {"wildcard":{"LABEL_G":{"value":"*货币*"}}}

[0170] }

[0171] 当非叶子节点对应的逻辑运算符为“NOT”时,根据逻辑运算符“NOT”对应装配规则和与非叶子节点关联的至少两个叶子节点分别对应的子句获得的第二ES查询子句为:

[0172]
```
{"bool": {
    "must_not": [
        {"term": {"LABEL_E": "20150101"}},
        {"range": {"LABEL_F": {"lte": "20231231"}}}
    ]
}
```

[0173] 第四,根据第一ES查询子句和至少一个非叶子节点分别对应的第二ES查询子句按照树形数据结构自顶向下顺序组装获得ES查询语句。

[0174] 进一步地,结合图3B,自顶向下依次组装,得到的ES查询语句为:

[0175]

```
{
  "query": {
    "bool": {
      "filter": {
        "bool": {
          "must": [
            { "term": {"LABEL_A": "value"} },
            { "terms": {"LABEL_B": ["T1","T2"]} },
            {
              "bool": {
                "should": [
                  {"range": {"LABEL_C": {"gte": 10}}},
                  {"term": {"LABEL_D": 3}},
                  {
                    "bool": {
                      "must_not": [
                        {"term": {"LABEL_E": "20150101"}},
                        {"range": {"LABEL_F": {"lte": "20231231"}}}
                      ]
                    }
                  }
```

```
                    ],
                    "minimum_should_match": 1
                }
            },
            {
                "bool": {
                    "must": [
                        {"match": {"PD_NAME": "零钱"}},
                        {"wildcard": {"LABEL_G": {"value": "*货币*"}}}
                    ]
                }
            }
        ]
    }
  }
 }
}
```

[0176]

[0177] 进一步地,在根节点对应的逻辑运算符与第一非叶子节点对应逻辑运算符相同,且第一非叶子节点与根节点连接时,将第一ES查询子句与第一非叶子节点对应的第二ES查询子句进行合并处理。

[0178] 例如,在上述根据图3B获得的ES查询语句中,根节点对应的逻辑运算符为“AND”,存在一个非叶子节点对应逻辑运算符也为“AND”,且该非叶子节点与根节点相连,那么可以将根节点对应的第一ES查询子句和该非叶子节点对应的第二ES查询子句进行合并,合并后的结果为:

```
{"bool": {
    "must": [
        { "term": {"LABEL_A": "value"} },
```

[0179]

[0180] {"terms":{"LABEL_B":["T1","T2"]}},

[0181] {"match":{"PD_NAME":"零钱"}},

[0182] {"wildcard":{"LABEL_G":{"value":"*货币*"}}}

[0183] }

[0184] 当服务器接收到传送的复杂查询请求时,这种合并查询方式,可以减少查询的深度,简化查询结构,从而提高查询效率。

[0185] 步骤130:服务器根据ES查询语句访问ES服务,获得查询结果,查询结果指示满足逻辑条件集合的产品。具体地,将构建的ES查询语句通过超文本传输协议(HyperText Transfer Protocol,HTTP)请求发送到ES服务的应用编程接口端口(Application Program Interface,API),通常为RESTful API调用。ES服务接收到查询请求后,根据提供的ES查询语句查找匹配的产品的信息,并将查询到的结果返回,返回结果指示匹配ES查询语句的产品的信息。

[0186] 综上,用户可以在可视化界面上灵活定义逻辑条件,形成逻辑条件集合,用户只需进行点击选择操作,无需具备技术背景或了解ES查询语法,简化了查询流程,降低了技术门槛;将逻辑条件集合转换为树形数据结构,并按自上向下的顺序依次组装为相应的ES查询语句,实现了对这个树形数据结构自上向下的全面解析,确保了查询语句的结构完整性,减少了误差和遗漏;通过构建指标名称和ES字段信息的映射关系,能够实现快速确定叶子节点对应的子句,从而构建第一ES查询子句和第二ES查询子句,从而获得ES查询语句,进一步地,根据ES查询语句为用户快速精准搜索筛选结果,可以有效解决在复杂查询场景中现有技术存在查询精度低的问题。

[0187] 图4和图5为本申请的实施例提供的可能的搜索装置的结构示意图。这些搜索装置可以用于实现上述方法实施例中服务器的功能。

[0188] 如图4所示,搜索装置400包括通信单元410和处理单元420。

[0189] 当搜索装置400用于实现图1所示的方法实施例中服务器的功能时:

[0190] 所述通信单元410,用于接收用户操作指令;所述用户操作指令指示用户请求查询的产品所需满足的逻辑条件集合,所述逻辑条件集合包括多个逻辑条件,所述多个逻辑条件属于不同层级,每个逻辑条件包括一个逻辑运算符和至少两个指标的描述信息;

[0191] 所述处理单元420,用于根据所述逻辑条件集合确定树形数据结构;其中,所述树形数据结构包括一个根节点、至少一个非叶子节点、多个叶子节点,其中,所述根节点和所述非叶子节点对应逻辑运算符,所述叶子节点对应指标的描述信息;根据所述树形数据结构构建ES查询语句;根据所述ES查询语句访问ES服务,获得查询结果,所述查询结果指示满足所述逻辑条件集合的产品。

[0192] 在一种可能的设计中,所述树形数据结构中,所述根节点与至少两个叶子节点具有关联关系、所述根节点与至少一个非叶子节点连接,每个非叶子节点与至少两个叶子节点具有关联关系;

[0193] 所述处理单元420在根据所述树形数据结构构建ES查询语句时,具体用于:

[0194] 确定每个叶子节点对应的子句;

[0195] 根据所述根节点对应的逻辑运算符和与所述根节点关联的至少两个叶子节点分别对应的子句确定第一ES查询子句;

[0196] 根据每个非叶子节点对应的逻辑运算符和与所述非叶子节点关联的至少两个叶子节点分别对应的子句确定第二ES查询子句;

[0197] 根据所述第一ES查询子句和所述至少一个非叶子节点分别对应的第二ES查询子

句按照所述树形数据结构自顶向下顺序组装获得所述ES查询语句。

[0198] 每个指标的描述信息包括指标的名称、指标的筛选方式和相应的筛选值;所述处理单元420在确定每个叶子节点对应的子句时,具体用于:

[0199] 根据映射关系和每个叶子节点对应的指标的描述信息中的所述指标的名称确定相应的ES字段信息;所述映射关系指示N个指标名称与N个ES字段信息一一对应,N为正整数;

[0200] 根据每个叶子节点对应的指标的描述信息中的筛选方式确定相应的装配规则;

[0201] 根据获得的ES字段信息、所述装配规则以及所述筛选值确定该叶子节点对应的子句。

[0202] 所述映射关系根据产品信息集合构建,所述N个指标名称根据所述产品信息集合确定。

[0203] 所述筛选方式包括如下任意一项:等值匹配、多值等值匹配、不等式匹配、模糊匹配、搜索匹配。

[0204] 所述处理单元420,还用于:

[0205] 在所述根节点对应的逻辑运算符与第一非叶子节点对应逻辑运算符相同,且所述第一非叶子节点与所述根节点连接时,将所述第一ES查询子句与所述第一非叶子节点对应的第二ES查询子句进行合并处理。

[0206] 有关上述通信单元410和处理单元420更详细的描述可以直接参考图1所示的方法实施例中相关描述直接得到,这里不加赘述。

[0207] 如图5所示,搜索装置500包括处理器510和通信接口520。处理器510和通信接口520之间相互耦合。可以理解的是,通信接口520可以为收发器或输入输出接口。可选的,搜索装置500还可以包括存储器530,用于存储处理器510执行的指令或存储处理器510运行指令所需要的输入数据或存储处理器510运行指令后产生的数据。

[0208] 当搜索装置500用于实现图1所示的方法时,处理器510用于实现上述处理单元420的功能,通信接口520用于实现上述通信单元410的功能。

[0209] 本申请实施例中对单元的划分是示意性的,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,另外,在本申请各个实施例中的各功能单元可以集成在一个处理器中,也可以是单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0210] 基于同一发明构思,本申请实施例提供一种计算机可读存储介质,计算机程序产品包括:计算机程序代码,当计算机程序代码在计算机上运行时,使得计算机执行如前文论述任一的搜索方法。由于上述计算机可读存储介质解决问题的原理与搜索方法相似,因此上述计算机可读存储介质的实施可以参见方法的实施,重复之处不再赘述。

[0211] 基于同一发明构思,本申请实施例还提供了一种计算机程序产品,该计算机程序产品包括:计算机程序代码,当该计算机程序代码在计算机上运行时,使得计算机执行如前文论述任一的搜索方法。由于上述计算机程序产品解决问题的原理与搜索方法相似,因此上述计算机程序产品的实施可以参见方法的实施,重复之处不再赘述。

[0212] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实

施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0213] 本申请是参照根据本申请的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0214] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0215] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列用户操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0216] 尽管已描述了本申请的优选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例作出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本申请范围的所有变更和修改。

[0217] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

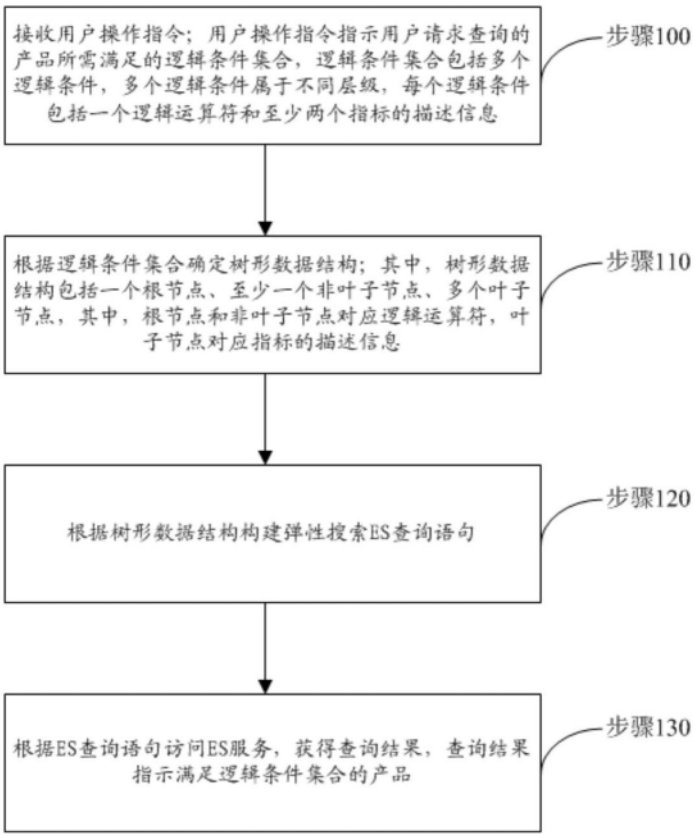

接收用户操作指令；用户操作指令指示用户请求查询的产品所需满足的逻辑条件集合，逻辑条件集合包括多个逻辑条件，多个逻辑条件属于不同层级，每个逻辑条件包括一个逻辑运算符和至少两个指标的描述信息
步骤100
根据逻辑条件集合确定树形数据结构；其中，树形数据结构包括一个根节点、至少一个非叶子节点、多个叶子节点，其中，根节点和非叶子节点对应逻辑运算符，叶子节点对应指标的描述信息
步骤110
根据树形数据结构构建弹性搜索ES查询语句
步骤120
根据ES查询语句访问ES服务，获得查询结果，查询结果指示满足逻辑条件集合的产品
步骤130

图1

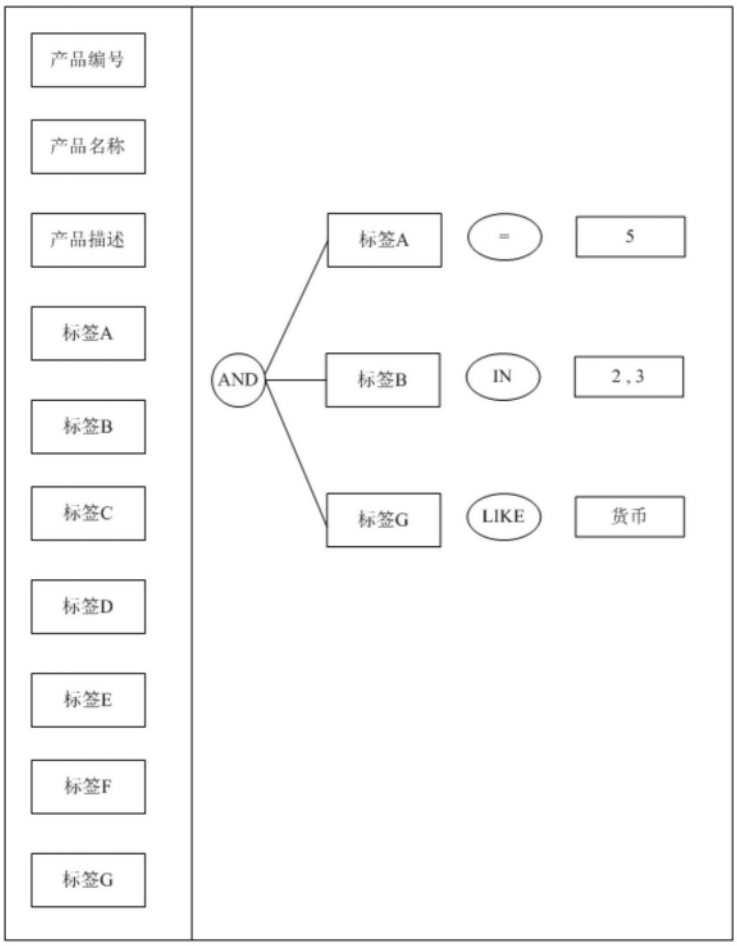

产品编号
产品名称
产品描述
标签A
标签B
标签C
标签D
标签E
标签F
标签G
AND
标签A
=
5
标签B
IN
2，3
标签G
LIKE
货币

图2

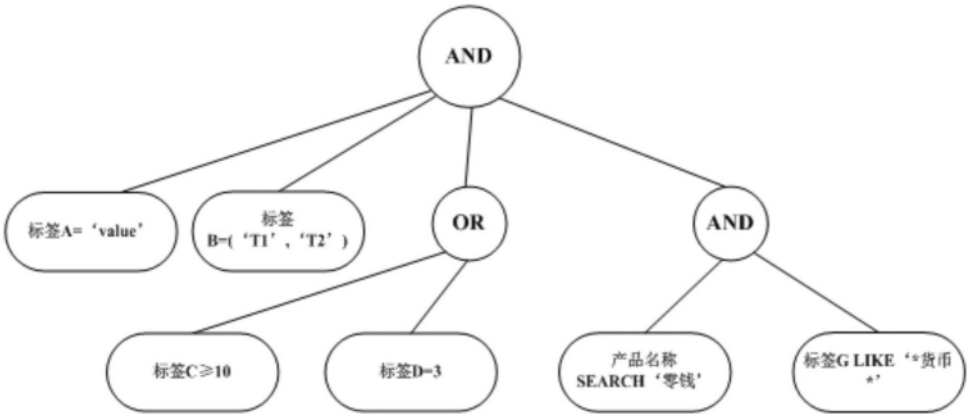

AND
标签A=‘value’
标签
B=(‘T1’，‘T2’)
OR
AND
标签C≥10
标签D=3
产品名称
SEARCH‘零钱’
标签G LIKE‘*货币*’

图3A

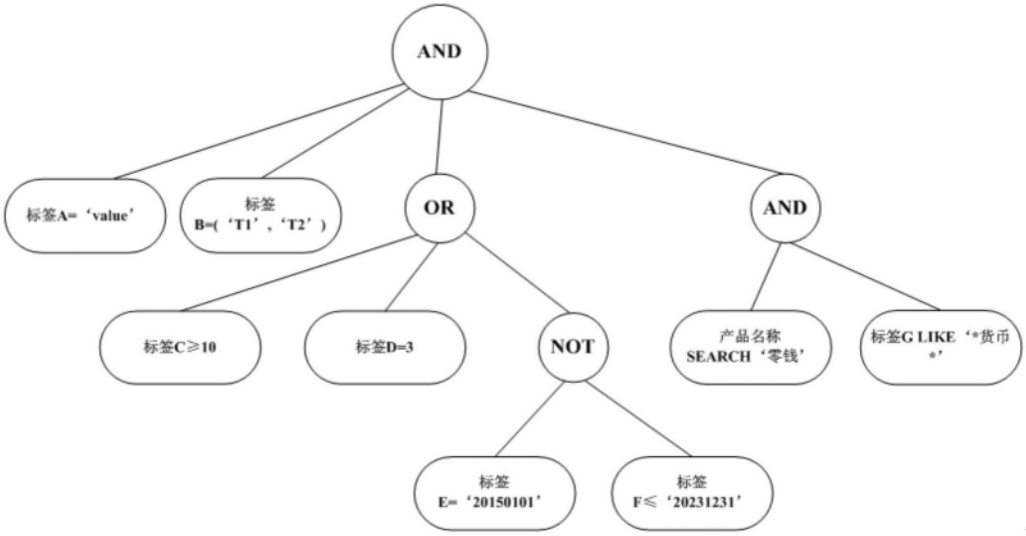

AND
标签A=‘value’
标签
B=(‘T1’,‘T2’)
OR
AND
标签C≥10
标签D=3
NOT
产品名称
SEARCH‘零钱’
标签G LIKE‘*货币*’
标签
E=‘20150101’
标签
F≤‘20231231’

图3B


搜索装置400
通信单元410
处理单元420

图4

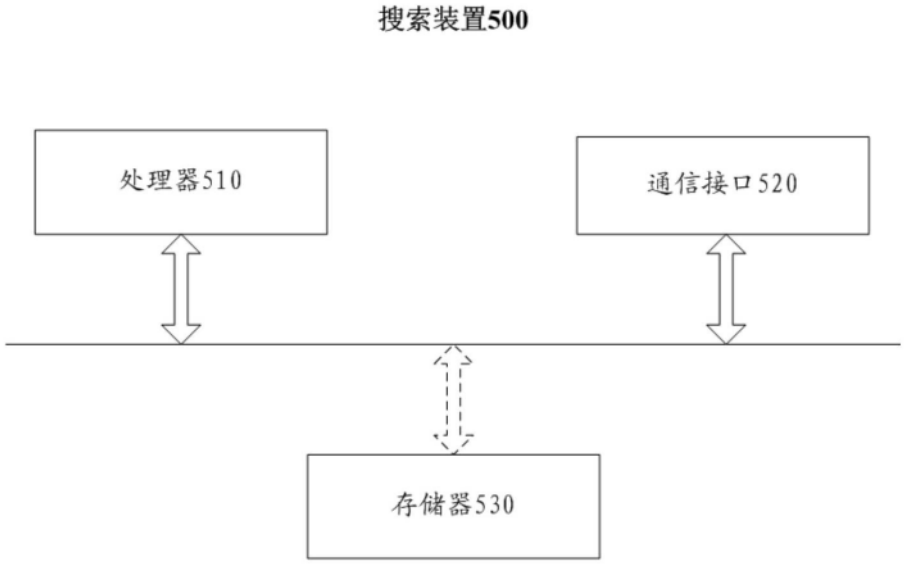


图5