

## 一种图数据的查询方法、装置以及设备

### 技术领域

[01]本说明书涉及大数据技术领域，尤其涉及一种图数据的查询方法、装置以及设备。

### 背景技术

[02]传统的数据库是关系型数据库，指采用了关系模型来组织数据的数据库，其以行和列的形式存储数据。而随着图计算技术，以及大数据技术和机器学习技术的迅速发展，图数据库也得到了越来越广泛的应用。图数据通常包含多个节点，以及节点之间的边，节点表示指定的对象，比如用户、业务特征等，边表示其连接的两个节点之间的关系，比如，两个节点之间发生过的交易、两个节点之间的好友或者亲属关系等。

[03]图数据库中的图数据查询是很重要的用户需求，用户经常需要了解图数据库中指定的图数据中包含哪些符合条件的节点（称为目标节点），比如，查询距离（一般指最短距离）指定点在指定跳数或者指定跳数以内的目标节点等。

[04]目前，在图数据库中逐跳进行查询时（比如，在 Neo4j 图数据库基于 Cypher 语句查询），会从指定的查找起始节点沿着边进行展开，获取查找起始节点的邻居节点，再获取该邻居节点的其他邻居节点，重复该步骤直到查找到目标节点，而在获取邻居节点时均会对当前的节点进行物化，从而会引入大量的物化过程，这里的物化指根据索引获取到内存或磁盘中存储的该指定向的节点的值信息（包括一些具体的业务属性值，比如，相关具体业务数据、邻居节点信息等），而物化的过程又比较耗时的。

[05]基于此，需要更为高效的图数据查询方案。

### 发明内容

[06]本说明书一个或多个实施例提供一种图数据的查询方法、装置、设备以及存储介质，用以解决如下技术问题：需要更为高效的图数据查询方案。

[07]为解决上述技术问题，本说明书一个或多个实施例是这样实现的。

[08]本说明书一个或多个实施例提供的一种图数据的查询方法，包括：接收针对图数据的查询请求；在所述图数据中确定查找起始节点；从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；根据响应于所述查

询请求所需返回的查询数据，判断是否需要与所述目标节点进行物化；若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

[09]本说明书一个或多个实施例提供一种图数据的查询装置，包括：请求接收模块，接收针对图数据的查询请求；起始确定模块，在所述图数据中确定查找起始节点；沿边查找模块，从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；物化判断模块，根据响应于所述查询请求所需返回的查询数据，判断是否需要与所述目标节点进行物化；结果返回模块，若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

[10]本说明书一个或多个实施例提供一种图数据的查询设备，包括：至少一个处理器；以及，与所述至少一个处理器通信连接的存储器；其中，所述存储器存储有可被所述至少一个处理器执行的指令，所述指令被所述至少一个处理器执行，以使所述至少一个处理器能够：接收针对图数据的查询请求；在所述图数据中确定查找起始节点；从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；根据响应于所述查询请求所需返回的查询数据，判断是否需要与所述目标节点进行物化；若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

[11]本说明书一个或多个实施例提供一种非易失性计算机存储介质，所述介质存储有计算机可执行指令，所述计算机可执行指令设置为：接收针对图数据的查询请求；在所述图数据中确定查找起始节点；从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；根据响应于所述查询请求所需返回的查询数据，判断是否需要与所述目标节点进行物化；若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

[12]本说明书一个或多个实施例采用的上述至少一个技术方案能够达到以下有益效果：主动根据查询请求所需返回的查询数据，来判断是否需要与目标节点进行物化，而不是按照目前采用的标准流程无脑地对所有目标节点和中间节点都进行物化，比如，在需要返回的查询数据依赖于目标节点的值信息时才考虑物化，而在诸如只需要返回目标节点的数量等不依赖于目标节点的值信息的查询数据时，则不对目标节点进行物化，从而，在目标节点较多的情况下，尤其能够节省大量的物化操作耗费的时间，从而提高了图数据查询效率；不仅如此，还可选地主动不对中间节点进行物化，从而可能节省更多的物化操作耗费的时间，更好地提高了图数据查询效率。

## 附图说明

[13]为了更清楚地说明本说明书实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本说明书中记载的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[14]图 1 为本说明书一个或多个实施例提供了一种图数据的查询方法的流程示意图；

[15]图 2 为本说明书一个或多个实施例提供了一种图数据的局部区域的示意图；

[16]图 3 为本说明书一个或多个实施例提供了一种应用场景下，图 1 中方法的一种具体实施方案的流程示意图；

[17]图 4 为本说明书一个或多个实施例提供了一种图数据的查询装置的结构示意图；

[18]图 5 为本说明书一个或多个实施例提供了一种图数据的查询设备的结构示意图。

## 具体实施方式

[19]本说明书实施例提供一种图数据的查询方法、装置、设备以及存储介质。

[20]为了使本技术领域的人员更好地理解本说明书中的技术方案，下面将结合本说明书实施例中的附图，对本说明书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。基于本说明书实施例，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，都应当属于本申请保护的范围。

[21]正如背景技术中提到的，在一些标准处理流程中，可能产生大量的耗费时间的物化（或者也可以叫具化）操作。关系型数据库也存在大量物化操作耗时严重的问题，本申请提供了相应的方案，需要注意的是，关系型数据库中的物化和图数据库中的物化是有区别的。

[22]对于图数据库，物化指根据索引获取到内存或者磁盘中存储的节点的值信息（包括节点的一些具体的属性值，比如，用户姓名、年龄地址、会员等级、交易数量、最近一笔订单等），节点为图数据中包含的节点。

[23]对于列式的关系型数据库，物化指将列式存储的数据转换成易于阅读理解的行形式，而在查询时若优先物化全部数据，就完全浪费了列式存储的优势，因此可以考虑采

用延迟物化策略，比如，直接在列式数据上进行过滤，然后借助位图对完成过滤的列式数据做位与操作，最后将结果和查询所需要返回的字段进行投影，再进行物化即可。

[24]对于行式的关系型数据库，物化指物化视图，即将满足某种条件的数据暂存为临时表以供后续使用，是一种对子查询的优化。则可以考虑通过合并子查询消除物化。

[25]但是，所考虑的上述方案是针对关系模型的，并不是适用于图模型，因此无法在图数据库中对图数据的查询中使用。因此，本申请进一步地再提供适用于图数据的查询方案，以解决大量物化操作影响图数据查询效率的问题。

[26]图 1 为本说明书一个或多个实施例提供的一种图数据的查询方法的流程示意图。该方法可以应用于不同的业务领域的图数据库中，这些业务领域比如包括：电子支付业务领域、电商业务领域、即时通讯业务领域、游戏业务领域、公务业务领域等。该流程可以在这些领域中存储有图数据的设备上执行，典型的如图数据库服务器等，流程中的某些输入参数或者中间结果允许人工干预调节，以帮助提高准确性。

[27]图 1 中的流程可以包括以下步骤 S102 至步骤 S110。

[28]S102：接收针对图数据的查询请求。

[29]在本说明书一个或多个实施例中，图数据包括多个节点，以及节点之间的边，节点表示指定的对象，比如用户、用户特征、业务特征、交易等，边表示其连接的两个节点之间的关系（大多是业务关系），比如，两个用户之间发生过的交易、两个用户之间的好友或者亲属关系等、两个业务特征之间的因果关系等。

[30]在本说明书一个或多个实施例中，查询请求用于请求查找：对应的图数据中第  $k$  层中与源节点相关联的所有节点，这里的第  $k$  层表示与源节点距离  $k$  跳（若两个节点之间有一条边连接，则该两个节点之间的距离为 1 跳）， $k$  为不小于 1 的正整数，将这类图数据查询称为  $k$ -hop 查询。具体比如，可以指定节点，作为查找起始节点，即源节点，使查询请求用于请求查找：距离查找起始节点指定跳数的节点的数量或者值信息，将距离指定节点指定跳数的节点作为目标节点，查询后返回目标节点的相关信息，作为对查询请求的响应结果。在  $k$ -hop 查询的情况下，目前尤其会涉及较多的物化操作，因此，本方案带来的效率提升效果较为明显，下面的一些实施例主要以  $k$ -hop 查询场景为例说明。当然，其他类型的查询，若涉及物化操作，同样可以考虑采用本方案来提升效率。

[31]S104：在所述图数据中确定查找起始节点。

[32]在本说明书一个或多个实施例中，查询请求中可以直接指定一个或者多个查找起始

节点。

[33]在没指定的情况下，可以自行挖掘确定查找起始节点，比如，假定想要在图数据中挖掘风险节点集合，则可以先尝试找到一些业务风险相对大的节点，然后将这些节点作为查找起始节点，查找其周围关系密切的节点，这些节点能够构成一个风险节点集合。

[34]S106：从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点。

[35]在本说明书一个或多个实施例中，从查找起始节点出发，逐跳展开向外侧游走，若同一个查找起始节点具有多条边，则分别从这各条边向外侧游走。在 k-hop 查询的情况下，则是通过向外侧游走，尝试查找距离 k 跳的节点，作为符合查询请求的目标节点，具体比如采用深度优先遍历的方式，遍历 k 跳以内的节点，从而找到各目标节点。

[36]S108：根据响应于所述查询请求所需返回的查询数据（即，至少部分查询结果），判断是否需要所述目标节点进行物化。

[37]响应于 k-hop 查询请求，在图数据中查找时，向外侧游走在到达目标节点之前，会经过一些中间节点。比如，图 2 为本说明书一个或多个实施例提供的一种图数据的局部区域的简化的示意图，圆圈表示节点，箭头边表示有向的边（该例中采用了有向图，而无向图也是可以的），边上标注的数字为对应的跳数，假定 k=5，A 为查找起始节点，需要向外侧移动 5 跳（若有多条可达路径，则可以以最短路径为准），到达目标节点，则途中会经过距离查找起始节点 1 跳（B 和 D）、2 跳（C 和 E）、3 跳（F）、4 跳（G）的节点，将这些节点称为中间节点，能够查找得到距离 A 为 5 跳的节点 H 和 I，作为目标节点。在标准流程下，在游走查找的过程中，会对这些中间节点和目标节点都进行物化，通过物化能够获取这些节点的值信息，基于此带来了大量耗费时间的物化操作。

[38]在本说明书一个或多个实施例中，在 k-hop 查询的情况下，需要返回的查询数据通常不包括中间节点的相关信息，物化中间节点得到的相关信息可能只作为他用或者缓存备用，因此，考虑对标准流程进行改进，不对所有的中间节点进行物化，以避免影响查询效率，若亟需这些相关信息时，再重启对中间节点的物化。

[39]而对于目标节点情况相对复杂一些。响应于查询请求需要返回目标节点的相关信息，但是具体是哪些信息则取决于用户实际需求，若无需物化目标节点即能获得所需返回的查询数据，则果断决策也不对目标节点进行物化。比如，若响应于查询请求所需返回的查询数据为目标节点的数量，在这种情况下，只需对目标节点计数就行而无需依赖于目

标节点的值信息，则可以判断不需要对目标节点进行物化。再比如，若响应于查询请求所需返回的查询数据为是否有目标节点命中，则只需返回判断结果即可，而不需要对目标节点进行物化。

[40]基于这样的思路，对于响应于查询请求所需返回的查询数据，根据查询数据是否依赖于目标节点的值信息，判断是否需要目标节点进行物化，若查询数据不依赖于目标节点的值信息，则判断的结果为否。可选地，反之，若查询数据依赖于目标节点的值信息（比如，需要直接返回值信息，或者需要对值信息进行计算后返回结果等），则判断的结果为是，当然在这一情况分支下也可以结合一些其他的条件，来最终决定判断结果，比如，若本地已缓存有该值信息，则虽然仍依赖，但也可不进行相应的物化操作。

[41]在本说明书一个或多个实施例中，在判断是否需要目标节点进行物化之后，若判断的结果为是，则对目标节点进行物化，得到目标节点的值信息，根据目标节点的值信息，返回相应的查询数据。这种情况下，需返回的查询数据是依赖于目标节点的值信息的，比如，需要返回的数据包括各目标节点的详细数据记录，则依赖于值信息。

[42]在实际应用中，查询数据是否依赖于目标节点的值信息的情况一般是有限枚举出来的，基于此，可以预设一些用于响应查询请求的一些查询数据类型，根据是否命中预设的这些类型，判断是否需要目标节点进行物化。如此，降低了对于依赖关系的分析难度，使得判断过程的执行效率更高。

[43]S110：若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

[44]通过图1的方法，主动根据查询请求所需返回的查询数据，来判断是否需要目标节点进行物化，而不是按照目前采用的标准流程无脑地对所有目标节点和中间节点都进行物化，比如，在需要返回的查询数据依赖于目标节点的值信息时才考虑物化，而在诸如只需要返回目标节点的数量等不依赖于目标节点的值信息的查询数据时，则不对目标节点进行物化，从而，在目标节点较多的情况下，尤其能够节省大量的物化操作耗费的时间，从而提高了图数据查询效率；不仅如此，还可选地主动不对中间节点进行物化，从而可能节省更多的物化操作耗费的时间，更好地提高了图数据查询效率。

[45]基于图1的方法，本说明书还提供了该方法的一些具体实施方案和扩展方案，下面继续进行说明。

[46]节点的值信息中可能会包含其邻居节点列表，在这种情况下，在图数据中查找目标节点时，可以利用各中间节点的邻居节点列表，来确定下一步的查找路径，这种用途使

得标准流程中的对中间节点的物化操作更加理所当然，虽然有损于效率但更倾向保留对中间节点的物化操作。

[47]为了扭转这种倾向，避免对中间节点进行物化，直接基于边来查找路径，包括：从查找起始节点出发，沿着图数据中的边在图数据中进行查找，根据在图数据中已到达的边，确定下一步的查找路径，而不对该边连接的中间节点进行物化。边上可能包含其连接的两个节点的标识，该标识用于唯一确定对应的节点，往往由开发人员根据自己公司的规则进行预定义，可以用作该节点对应的数据记录的主键，而不视为节点的值信息。

[48]而若将该标识其视为节点的值信息，那么需要对边也进行相应的物化操作以获取该标识，如此仍然不利于削减物化操作，考虑到这种情况。为了提高方案的适用性，进一步改造方案。具体包括：预先为图数据构建一个或者多个信息简化拓扑图，该信息简化拓扑图相比于原始的图数据，只包含尽量少的信息（其拓扑相比于图数据本身所表示的原始的拓扑也可以是不完整的），比如，为边关联的节点的标识；之后，在确定下一步的查找路径时，可以获取该信息简化拓扑图，据其确定在图数据中已到达的边所关联的对端节点的标识，从而能够确定下一步的查找路径。对于已确定的查找路径或者被排除的查找路径（比如，某些有回环趋势的路径，这些路径可能会导致跳数反而减小，而不是逐跳增加），尤其是其中一些决定路径正确与否的分歧处，可以进行标记，更新到信息简化拓扑图中，对于后续的其他查询可能能够复用，以加快查询速度。

[49]在本说明书一个或多个实施例中，图数据本身表示了一个拓扑图，称为原始的拓扑图，信息简化拓扑图则是根据该图数据构建的另一个拓扑图，信息简化拓扑图包含了该图数据中的局部区域，以用于快速查询。进一步地，根据多个不同的图数据来构建同一个信息简化拓扑图，则信息简化拓扑图包含了各图数据中的局部区域。不同的图数据的业务之间是可以有差异性的，他们分别包含的节点可以有一部分是相同的，由于不同的图数据关注的业务、功能之间的差异性，导致这些图数据所挖掘和表现出的节点以及节点间关系也会有区别，考虑以信息简化拓扑图作为中间媒介，使得原本相互隔离的不同的图数据之间，可以实现一定程度的数据交流，而无需全面融合使用，如此，既提高了数据的共享性，降低了不同图数据的业务盲点，又方便各个图数据对自己的私有数据进行管理和访问控制，从而也兼顾了安全性。

[50]基于这样的思路，具体比如，在确定下一步的查找路径时，若在当前的图数据（即S102中的图数据）中已到达的边能够延伸至信息简化拓扑图中，则在信息简化拓扑图中的局部区域（比如，对应于该图数据的局部区域）中确定下一步的查找路径。在信息

简化拓扑图包含多个不同的图数据中的局部区域的情况下,可以判断所查找的路径是否达到其所处的一个局部区域的边界处,假定该局部区域属于另一个图数据,若数据则可以进行图跳转,从该局部区域跳转至另一个图数据中的其他区域,继续确定下一步的查找路径,从而,有助于更全面可靠地找到各目标节点,突破了当前的图数据对应的业务所带来的关系挖掘方面的局限性。还可以根据当前的图数据与另一个图数据之间的关系,确定是否进行图跳转,从而能够更合理地进行相对有必要的图跳转,比如,根据该关系,确定是否符合访问控制策略,双方业务之前是否有关联性,预测双方是否均为发出查询请求的用户相关的图数据,等等。

[51]在实际执行时,路径的确定还和在图数据中的游走模式相关,比如,是基于深度优先遍历的模式进行游走,还是随机游走等。根据查询请求生成的查询计划树,查询计划树中能够提供相应的用于定义模式的配置属性,那么可以预先在查询计划树中定义游走模式,进而,根据上述的对端节点的标识和查询计划树,确定下一步的查找路径。这里,查询计划树是可选的,可以单独根据信息简化拓扑图,来确定下一步的查找路径。

[52]在本说明书一个或多个实施例中,虽然查询请求本身能够指示所需返回的查询数据,但是对于发送该查询请求的用户而言,对于所返回数据的具体形式未必有明确的期望,也即查询请求的指示未必与用户的真实期望相匹配(用户虽然下达了查询请求,但未必清楚将会怎样的形式返回结果)。基于此,本方案对用户的真实期望进行预测推断,以确定是否延迟进行物化操作,甚至不做物化操作,而不完全取决于查询请求的指示。如此,能够在满足用户真实期望的同时,尽量降低物化操作带来的不利影响,还能够对查询请求不适当的指示主动进行纠正。

[53]具体比如,无论 S108 的判断结果是什么,至少都可以在不对节点进行物化操作的情况下,先返回目标节点的实时数量(该过程可以是动态变化的,而不是等全部目标节点确定后再返回),作为最低限度的相应的查询数据;并且,可以检测查询请求对应的前端页面(即发送查询的用户那一端的交互页面)的页面实时状态,判断页面实时状态是否符合预定的用户专注条件,若是,则对于已返回相应的查询数据(最低限度的相应的查询数据)的目标节点,则对目标节点进行物化,并根据物化的结果,响应于查询请求进行补充返回。用户专注条件反映了:用户比较重视和专注于本次查询,或者,用户有较大概率期望返回更详细的查询数据,比如,包含目标节点的值信息的查询数据。通过检测页面实时状态,能够推理出用户在页面上即时的和/或很近一段时间内的操作和反映,比如,用户是否已经连续做了多次查询,用户多次查询之间的关联关系,是否密切

是否有程度的递进性，用户是否已在查询的前端页面上停留足够长的时间，用户是否主动复制过之前的详细查询结果，对于详细查询结果和简化查询结果用户浏览时间的差异程度，在多次查询的过程中，是否有无关页面反复切换至前端，等等。根据这些特征能够预测是否符合用户专注条件，进而确定是否主动补充基于物化的查询数据，哪怕是基于查询请求的指示，先前暂时判断了不进行物化，最终仍然延迟做了补充物化返回，从而能够提高用户体验，提高查询结果在用户体验方面的容错性。

[54]进一步地，由于延迟了基于物化结果的返回，因此增加了需要用户等待的空白时间，考虑主动填充这段空白时间，以避免用户被动地等待，使得整个查询以及结果返回的过程从用户角度来看更加地平滑、充实以及高效。该主动填充的操作最关注的是要让用户觉得这是理所当然的，且具有吸引力的，而不是冗余的等待，也即，虽然实质上延迟了，但是让用户不认为这是延迟。

[55]基于这样的思路，具体比如，对于已返回相应的查询数据（其对应的查询过程避免了物化操作，保证了效率）的目标节点，向所述前端页面推送反映沿边查找时相应的游走轨迹，并提示所述用户从中选择中间节点或者目标节点，以便在所述游走轨迹的展示过程中和所述用户的选择过程中，选择至少部分中间节点或者目标节点进行物化，若接收到用户的选择操作，则将相应的物化结果进行补充返回。在展示和选择过程中（此时，最低限度的查询结果已经返回完毕），可以并行地开始进行物化，以便在用户选择后，及时地筛选出匹配的补充结果返回，而若用户未进行选择，则可以将全部的物化结果都返回。这种方案，即符合用户的灵活可变的需求，又增强了用户的操控感，有助于用户阶段性地获得更精准而不冗余的查询结果，而且，还使得用户难以察觉到物化操作的延迟。兼顾了查询效率和用户对查询结果的真实期望。

[56]根据上面的说明，本说明书一个或多个实施例还提供了一种应用场景下，图 1 中方法的一种具体实施方案的流程示意图，如图 3 所示。在该场景下，基于 Cypher 语句进行 k-hop 查询，并按照上面的思路对查询的具体执行过程进行了改进。

[57]采用的 Cypher 语句的模板如下所示：

```
MATCH(a {})-->(b) RETURN count(DISTINCT b);
```

```
MATCH(a {})-->(b) RETURN sum(b.weight);
```

[58]第一条查询语句只需要返回目标节点的数量，但是，在改进前如背景技术所述仍需执行大量的物化操作。

[59]k-hop 查询经过优化，在图遍历时通过获取边的信息来寻找下一步路径，不断循环遍历直到将各目标节点都查找出来，在此过程中不对中间节点进行物化，而目标节点是否需要物化，取决于查询需要返回的数据，若需要返回的是目标节点的数量，比如上面的第一条查询语句，则比如通过计算边的数量获得到节点的数量，从而避免对目标节点的物化操作。但是，若需要返回的是目标节点的值信息，比如上面的第二条查询语句，需要返回节点的某个具体的属性值之和，则可以考虑对目标节点进行物化以获取属性值。同时，在考虑查询语句本身的基础上，对用户的真实期望进行预测，以决定如何延迟物化或者是否需要补充物化结果。

[60]图 3 中的流程可以包括以下步骤 S301 至步骤 S306。

[61]S301: 接收用户发送的针对图数据的查询请求。

[62]S302: 根据查询请求，生成相应的查询计划树，以用于指示查询过程执行，并从查询计划树中读取沿边游走查找时采用的遍历模式（比如，深度优先遍历），以及需返回的数据的类型。

[63]S303: 确定查找起始节点，根据读取的遍历模式，从查找起始节点发出沿边游走查找目标节点，过程中根据当前到达的边确定下一步的路径，而不对中间节点进行物化。

[64]S304: 判断需返回的数据的类型是目标节点的数量，还是目标节点具体的属性值，若是前者则执行 S305，若是后者则执行 S306。

[65]S305: 查找到目标节点后，将目标节点的实时数量返回，在返回后，可选地，检测查询请求对应的前端页面的页面实时状态，判断页面实时状态是否符合预定的用户专注条件，若是，则对目标节点进行物化以补充返回。

[66]S306: 查找到目标节点后，对目标节点进行物化，得到相应的属性值并返回，可选地，可以判断页面实时状态是否符合预定的用户专注条件，若否，则可以先不对目标节点进行物化，优先将目标节点的数量返回，之后再决策是否继续物化。

[67]通过图 3 中的方案，有效提高了 k-hop 查询的查询性能，一些测试数据反映在查询目标节点的数量量的情况下，耗时缩短为之前的五分之一。

[68]基于同样的思路，本说明书一个或多个实施例还提供了上述方法对应的装置和设备，如图 4、图 5 所示，虚线方框表示可选的模块。

[69]图 4 为本说明书一个或多个实施例提供的一种图数据的查询装置的结构示意图，所

述装置包括：请求接收模块 402，接收针对图数据的查询请求；起始确定模块 404，在所述图数据中确定查找起始节点；沿边查找模块 406，从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；物化判断模块 408，根据响应于所述查询请求所需返回的查询数据，判断是否需要所述目标节点进行物化；结果返回模块 410，若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

[70]可选地，还包括：物化执行模块 412；所述物化执行模块 412，在所述判断是否需要所述目标节点进行物化之后，若所述判断的结果为是，则对所述目标节点进行物化，得到所述目标节点的值信息；所述结果返回模块 410，根据所述目标节点的值信息，返回相应的查询数据。

[71]可选地，所述物化判断模块 408，对于响应于所述查询请求所需返回的查询数据，根据所述查询数据是否依赖于所述目标节点的值信息，判断是否需要所述目标节点进行物化；若所述查询数据不依赖于所述目标节点的值信息，则所述判断的结果为否。

[72]可选地，所述物化判断模块 408，若响应于所述查询请求所需返回的查询数据为所述目标节点的数量，则判断不需要对所述目标节点进行物化。

[73]可选地，所述沿边查找模块 406，从所述查找起始节点出发，沿着所述图数据中的边在所述图数据中进行深度优先查找；根据在所述图数据中已到达的边，确定下一步的查找路径，而不对所述边连接的中间节点进行物化，所述中间节点不为所述目标节点。

[74]可选地，所述沿边查找模块 406，获取预先根据所述图数据构建的信息简化拓扑图，在所述信息简化拓扑图中包含节点的标识；根据所述信息简化拓扑图，确定在所述图数据中已到达的边所关联的对端节点的标识；确定根据所述查询请求生成的查询计划树；根据所述对端节点的标识和所述查询计划树，确定下一步的查找路径。

[75]可选地，所述结果返回模块 410，在确定各目标节点的过程中，动态地返回已确定的目标节点的实时数量，作为相应的查询数据；检测所述查询请求对应的前端页面的页面实时状态，判断所述页面实时状态是否符合预定的用户专注条件；若是，则对于已返回相应的查询数据的目标节点，对所述目标节点进行物化，并根据所述物化的结果，响应于所述查询请求进行补充返回。

[76]可选地，所述结果返回模块 410，对于已返回相应的查询数据的目标节点，向所述前端页面推送反映沿边查找相应的游走轨迹，并提示所述用户从中选择中间节点或者目

标节点，以便在所述游走轨迹的展示过程中和所述用户的选择过程中，选择至少部分中间节点或者目标节点进行物化；若接收到用户的选择操作，则将相应的物化结果进行补充返回。

[77]可选地，所述查询请求中指定了所述查找起始节点；所述查询请求用于请求查找：距离所述查找起始节点指定跳数的节点的数量或者值信息，所述距离所述查找起始节点指定跳数的节点作为所述目标节点。

[78]可选地，所述图数据中的节点表示以下至少一种对象：用户、用户特征、业务特征、交易。

[79]图 5 为本说明书一个或多个实施例提供的一种图数据的查询设备的结构示意图，所述设备包括：至少一个处理器；以及，与所述至少一个处理器通信连接的存储器；其中，

[80]所述存储器存储有可被所述至少一个处理器执行的指令，所述指令被所述至少一个处理器执行，以使所述至少一个处理器能够：接收针对图数据的查询请求；在所述图数据中确定查找起始节点；从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；根据响应于所述查询请求所需返回的查询数据，判断是否需要对所述目标节点进行物化；若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

[81]处理器与存储器之间可以通过总线通信，设备还可以包括与其他设备通信的输入/输出接口。

[82]基于同样的思路，本说明书一个或多个实施例还提供了对应于图 1 中方法的一种非易失性计算机存储介质，存储有计算机可执行指令，所述计算机可执行指令设置为：接收针对图数据的查询请求；在所述图数据中确定查找起始节点；从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；根据响应于所述查询请求所需返回的查询数据，判断是否需要对所述目标节点进行物化；若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

[83]在 20 世纪 90 年代，对于一个技术的改进可以很明显地区分是硬件上的改进（例如，对二极管、晶体管、开关等电路结构的改进）还是软件上的改进（对于方法流程的改进）。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，可编程

逻辑器件（Programmable Logic Device, PLD）（例如现场可编程门阵列（Field Programmable Gate Array, FPGA））就是这样一种集成电路，其逻辑功能由用户对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片 PLD 上，而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且，如今，取代手工地制作集成电路芯片，这种编程也多半改用“逻辑编译器（logic compiler）”软件来实现，它与程序开发撰写时所用的软件编译器相类似，而要编译之前的原始代码也得用特定的编程语言来撰写，此称之为硬件描述语言（Hardware Description Language, HDL），而 HDL 也并非仅有一种，而是有许多种，如 ABEL（Advanced Boolean Expression Language）、AHDL（Altera Hardware Description Language）、Confluence、CUPL（Cornell University Programming Language）、HDCal、JHDL（Java Hardware Description Language）、Lava、Lola、MyHDL、PALASM、RHDH（Ruby Hardware Description Language）等，目前最普遍使用的是 VHDL（Very-High-Speed Integrated Circuit Hardware Description Language）与 Verilog。本领域技术人员也应该清楚，只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中，就可以很容易得到实现该逻辑方法流程的硬件电路。

[84]控制器可以按任何适当的方式实现，例如，控制器可以采取例如微处理器或处理器以及存储可由该（微）处理器执行的计算机可读程序代码（例如软件或固件）的计算机可读介质、逻辑门、开关、专用集成电路（Application Specific Integrated Circuit, ASIC）、可编程逻辑控制器和嵌入微控制器的形式，控制器的例子包括但不限于以下微控制器：ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及 Silicone Labs C8051F320，存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道，除了以纯计算机可读程序代码方式实现控制器以外，完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件，而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至，可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

[85]上述实施例阐明的系统、装置、模块或单元，具体可以由计算机芯片或实体实现，或者由具有某种功能的产品来实现。一种典型的实现设备为计算机。具体的，计算机例如可以为个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任何设备的组合。

[86]为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本说明书时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

[87]本领域内的技术人员应明白，本说明书实施例可提供为方法、系统、或计算机程序产品。因此，本说明书实施例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，本说明书实施例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。

[88]本说明书是参照根据本说明书实施例的方法、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。

[89]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。

[90]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。

[91]在一个典型的配置中，计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[92]内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

[93]计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。

计算机的存储介质的例子包括,但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带,磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括暂存电脑可读媒体(transitory media),如调制的数据信号和载波。

[94]还需要说明的是,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

[95]本说明书可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本说明书,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[96]本说明书中的各个实施例均采用递进的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于装置、设备、非易失性计算机存储介质实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[97]上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下,在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外,在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中,多任务处理和并行处理也是可以的或者可能是有利的。

[98]以上所述仅为本说明书的一个或多个实施例而已,并不用于限制本说明书。对于本领域技术人员来说,本说明书的一个或多个实施例可以有各种更改和变化。凡在本说明书的一个或多个实施例的精神和原理之内所作的任何修改、等同替换、改进等,均应包含在本说明书的权利要求范围之内。

## 权利要求书

1、一种图数据的查询方法，包括：

接收针对图数据的查询请求；

在所述图数据中确定查找起始节点；

从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；

根据响应于所述查询请求所需返回的查询数据，判断是否需要与所述目标节点进行物化；

若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

2、如权利要求 1 所述的方法，所述判断是否需要与所述目标节点进行物化之后，所述方法还包括：

若所述判断的结果为是，则对所述目标节点进行物化，得到所述目标节点的值信息；

根据所述目标节点的值信息，返回相应的查询数据。

3、如权利要求 1 所述的方法，所述根据响应于所述查询请求所需返回的查询数据，判断是否需要与所述目标节点进行物化，包括：

对于响应于所述查询请求所需返回的查询数据，根据所述查询数据是否依赖于所述目标节点的值信息，判断是否需要与所述目标节点进行物化；

若所述查询数据不依赖于所述目标节点的值信息，则所述判断的结果为否。

4、如权利要求 1 所述的方法，所述根据响应于所述查询请求所需返回的查询数据，判断是否需要与所述目标节点进行物化，包括：

若响应于所述查询请求所需返回的查询数据为所述目标节点的数量，则判断不需要与所述目标节点进行物化。

5、如权利要求 1 所述的方法，所述从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点，包括：

从所述查找起始节点出发，沿着所述图数据中的边在所述图数据中进行查找；

根据在所述图数据中已到达的边，确定下一步的查找路径，而不对所述边连接的中间节点进行物化，所述中间节点不为所述目标节点。

6、如权利要求 5 所述的方法，所述根据在所述图数据中已到达的边，确定下一步的查找路径，包括：

获取预先根据所述图数据构建的信息简化拓扑图，在所述信息简化拓扑图中包含节点的标识；

根据所述信息简化拓扑图，确定在所述图数据中已到达的边所关联的对端节点的标识；

确定根据所述查询请求生成的查询计划树；

根据所述对端节点的标识和所述查询计划树，确定下一步的查找路径。

7、如权利要求 6 所述的方法，所述信息简化拓扑图与所述图数据所表示的原始的拓扑图一部分一致，所述信息简化拓扑图根据多个不同的所述图数据而构建，所述信息简化拓扑图包含了各所述图数据中的局部区域；

所述确定下一步的查找路径，包括：

若所述在所述图数据中已到达的边能够延伸至所述信息简化拓扑图中，则在所述信息简化拓扑图中的所述局部区域中确定下一步的查找路径。

8、如权利要求 7 所述的方法，所述在所述信息简化拓扑图中的所述局部区域中确定下一步的查找路径，包括：

判断所查找的路径是否达到其所处的一个所述局部区域的边界处，该局部区域属于另一个图数据；

若属于，则根据当前的图数据与所述另一个图数据之间的关系，确定是否进行图跳转；

若是，则从所述该局部区域跳转至所述另一个图数据中的其他区域，继续确定下一步的查找路径。

9、如权利要求 1 所述的方法，所述在不对所述目标节点进行物化的情况下，返回相应的查询数据，包括：

在确定各目标节点的过程中，动态地返回已确定的目标节点的实时数量，作为相应的查询数据；

所述方法还包括：

检测所述查询请求对应的前端页面的页面实时状态，判断所述页面实时状态是否符合预定的用户专注条件；

若是，则对于已返回相应的查询数据的目标节点，对所述目标节点进行物化，并根据所述物化的结果，响应于所述查询请求进行补充返回。

10、如权利要求 9 所述的方法，所述对于已返回相应的查询数据的目标节点，对所述目标节点进行物化，包括：

对于已返回相应的查询数据的目标节点，向所述前端页面推送反映沿边查找时相应的游走轨迹，并提示所述用户从中选择中间节点或者目标节点，以便在所述游走轨迹的

展示过程中和所述用户的选择过程中，选择至少部分中间节点或者目标节点进行物化；  
若接收到用户的选择操作，则将相应的物化结果进行补充返回。

11、如权利要求 1~10 任一项所述的方法，所述查询请求中指定了所述查找起始节点；

所述查询请求用于请求查找：距离所述查找起始节点指定跳数的节点的数量或者值信息，所述距离所述查找起始节点指定跳数的节点作为所述目标节点。

12、如权利要求 1~10 任一项所述的方法，所述图数据中的节点表示以下至少一种对象：用户、用户特征、业务特征、交易。

13、一种图数据的查询装置，包括：

请求接收模块，接收针对图数据的查询请求；

起始确定模块，在所述图数据中确定查找起始节点；

沿边查找模块，从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；

物化判断模块，根据响应于所述查询请求所需返回的查询数据，判断是否需要对所  
述目标节点进行物化；

结果返回模块，若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。

14、如权利要求 13 所述的装置，还包括：物化执行模块；

所述物化执行模块，在所述判断是否需要所述目标节点进行物化之后，若所述判断的结果为是，则对所述目标节点进行物化，得到所述目标节点的值信息；

所述结果返回模块，根据所述目标节点的值信息，返回相应的查询数据。

15、如权利要求 13 所述的装置，所述物化判断模块，对于响应于所述查询请求所需返回的查询数据，根据所述查询数据是否依赖于所述目标节点的值信息，判断是否需要所述目标节点进行物化；

若所述查询数据不依赖于所述目标节点的值信息，则所述判断的结果为否。

16、如权利要求 13 所述的装置，所述物化判断模块，若响应于所述查询请求所需返回的查询数据为所述目标节点的数量，则判断不需要对所述目标节点进行物化。

17、如权利要求 13 所述的装置，所述沿边查找模块，从所述查找起始节点出发，沿着所述图数据中的边在所述图数据中进行查找；

根据在所述图数据中已到达的边，确定下一步的查找路径，而不对所述边连接的中间节点进行物化，所述中间节点不为所述目标节点。

18、如权利要求 17 所述的装置，所述沿边查找模块，获取预先根据所述图数据构建的信息简化拓扑图，在所述信息简化拓扑图中包含节点的标识；

根据所述信息简化拓扑图，确定在所述图数据中已到达的边所关联的对端节点的标识；

确定根据所述查询请求生成的查询计划树；

根据所述对端节点的标识和所述查询计划树，确定下一步的查找路径。

19、如权利要求 13~18 任一项所述的装置，所述查询请求中指定了所述查找起始节点；

所述查询请求用于请求查找：距离所述查找起始节点指定跳数的节点的数量或者值信息，所述距离所述查找起始节点指定跳数的节点作为所述目标节点。

20、如权利要求 13~18 任一项所述的装置，所述图数据中的节点表示以下至少一种对象：用户、用户特征、业务特征、交易。

21、一种图数据的查询设备，包括：

至少一个处理器；以及，

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述至少一个处理器执行的指令，所述指令被所述至少一个处理器执行，以使所述至少一个处理器能够：

接收针对图数据的查询请求；

在所述图数据中确定查找起始节点；

从所述查找起始节点出发，沿着所述图数据中的边，在所述图数据中查找符合所述查询请求的目标节点；

根据响应于所述查询请求所需返回的查询数据，判断是否需要所述目标节点进行物化；

若否，则在不对所述目标节点进行物化的情况下，返回相应的查询数据。