(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
05 April 2018 (05.04.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/064241 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/US2017/053855

(22) International Filing Date:
28 September 2017 (28.09.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/282,955 30 September 2016 (30.09.2016) US

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).

(72) Inventors: **SARKAR, Angshuman**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, WA 98052-6399 (US). **NIELSEN, Henrik, Frystyk**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, WA 98052-6399 (US). **TAYLOR, John, Anthony**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, WA 98052-6399 (US).

(74) Agent: **MINHAS, Sandip, S.** et al.; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, WA 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: DISAMBIGUATING COLLECTION ACCESS IN A GRAPH DATABASE, SUBJECT-PREDICATE-OBJECT DATABASE, AND/OR TRIPLESTORE

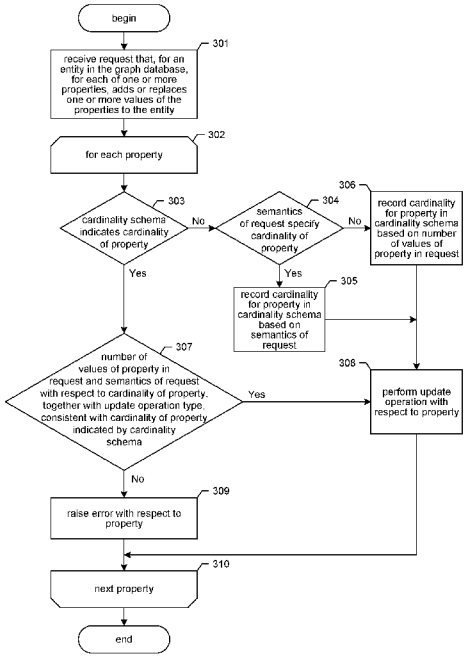

FIG. 3

(57) Abstract: A facility maintains a cardinality schema to manage indications of property cardinality on behalf of a database such as a graph database that is indifferent to the cardinality of properties stored within it. The facility can add cardinality indications to the cardinality schema for a property based on (1) semantics used to refer to one or more values of the property in an update request, (2) the number of values specified for the property in an update request, or (3) an explicit client declaration of the property's cardinality. Where a query or update request against the database uses cardinality semantics for a property that are inconsistent with the cardinality schema's indication for the property, the facility can reject the query or update request. The facility can also use the cardinality schema to automatically complete incomplete queries using the proper cardinality semantics for each involved property.

WO 2018/064241 A1

[Continued on next page]

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

DISAMBIGUATING COLLECTION ACCESS IN A GRAPH DATABASE, SUBJECT-PREDICATE-OBJECT DATABASE, AND/OR TRIPLESTORE

BACKGROUND

[0001] Triplestores and subject-predicate-object databases store "triples" each representing a relationship between a pair of things. For example, such a database may contain the triple (Sandra, manages, Anthony). In this example, "Sandra" is the subject of the triple, "manages" is the predicate of the triple, and "Anthony" is the object of the triple. The triple represents a relationship between Sandra and Anthony in which Sandra is Anthony's manager.

[0002] Graph databases organize information by connecting pairs of nodes each with an edge. A graph database can be used to represent the contents of a triplestore or subject-predicate-object database; in particular, each triple is represented by establishing a first node corresponding to the subject of the triple, establishing a second node corresponding to the object of the triple, and establishing an edge from the first node to the second node corresponding to the predicate of the triple. In some cases, graph databases are used to represent the values of properties for entities: a first node represents an entity, such as a particular book; an edge exiting the first node represents a property identity, such as a title property; and a second node entered by the edge represents a value of the property represented by the edge for the entity represented by the first node, such as the title of the book represented by the first node.

[0003] Graph databases, subject-predicate-object databases, and triplestores have many similarities; can be used in many of the same situations; and can often be straightforwardly transformed between each other. They are favored for their flexibility, able to dynamically consume and organize data of arbitrary complexity without requiring that a structure be defined for the data in advance of its loading.

SUMMARY

[0004] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to identify key factors or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0005] A facility maintains a cardinality schema to manage indications of property cardinality on behalf of a database such as a graph database that is indifferent to the cardinality of properties stored within it. The facility can add cardinality indications to the

cardinality schema for a property based on (1) semantics used to refer to one or more values of the property in an update request, (2) the number of values specified for the property in an update request, or (3) an explicit client declaration of the property's cardinality. Where a query or update request against the database uses cardinality semantics for a property that are inconsistent with the cardinality schema's indication for the property, the facility can reject the query or update request. The facility can also use the cardinality schema to automatically complete incomplete queries using the proper cardinality semantics for each involved property.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Figure 1 is a block diagram showing some of the components typically incorporated in at least some of the computer systems and other devices on which the facility operates.

[0007] Figure 2 is a storage diagram showing data and code stored by the facility in some embodiments.

[0008] Figure 3 is a flow diagram showing a process performed by the facility in some embodiments to process update requests.

[0009] Figure 4 is a table diagram showing logic used by the facility in some embodiments to perform act 305.

[0010] Figures 5 and 6 are table diagrams showing logic used by the facility in some embodiments to perform act 307.

[0011] Figure 7 is a graph diagram showing the contents of the database corresponding to Table 1C in graph form.

[0012] Figure 8 is a flow diagram showing a process performed by the facility in some embodiments to alter the cardinality schema in response to an explicit request from a client application.

[0013] Figure 9 is a flow diagram showing a process performed by the facility in some embodiments to respond to a query against the database.

[0014] Figures 10-12 are table diagrams showing logic used by the facility in some embodiments to perform acts 905 and 907.

[0015] Figure 13 is a flow diagram showing a process performed by the facility in some embodiments to automatically complete incomplete queries using proper cardinality semantics for the involved properties.

[0016] Figure 14 is a display diagram showing a sample display depicting the entry of an incomplete query.

[0017] Figure 15 is a display diagram showing a sample display presented by the facility in some embodiments to propose a constructed predictive full query as a substitute for an entered partial query

DETAILED DESCRIPTION

[0018] The inventors have recognized that the flexibility of graph databases, subject-predicate-object databases, and triplestores noted above is in many cases mismatched with data updating mechanisms and data querying mechanisms commonly used with these databases, which impose greater discipline on aspects of datatyping, including the notion of "cardinality."

[0019] Cardinality, as used herein, refers to the number of values that an entity can have for a particular property. For example, in a database in which the property "social security number" can have at most one value for each person, the "social security number" property is referred to as having a "scalar," or "single-value" cardinality level. In a database in which the property "telephone number" can have multiple values for a single person, the "telephone number" property is referred to as having a "collection," "multiple-value," or "array" cardinality level.

[0020] While graph databases, subject-predicate-object databases, and triplestores typically do not discern or store cardinalities for properties or predicates, data updating mechanisms and data querying mechanisms commonly used with these databases are typically aware of the cardinality of properties and predicates, and often include semantics in update requests and queries--optional or mandatory--that identify properties or predicates included in the update requests and queries as having a particular cardinality level. The inventors have recognized that the typical failure of graph databases, subject-predicate-object databases, and triplestores to track the cardinality of the properties and predicates that they store interferes with their ability to respond appropriately to some expressions of cardinality included in the semantics of update requests and queries these databases receive. In particular, these mechanisms may seek to update or query on a property or predicate that should be a scalar as a collection, or vice versa. When the database treats these incorrect cardinality levels as if they are correct, they deprive the clients accessing the data through these mechanisms of information about the correct cardinality of a property or predicate, making it more likely that the data will become corrupted with respect to the proper cardinality of properties or predicates, and that the results of queries will be misinterpreted.

[0021] To ameliorate the disadvantages associated with this mismatch observed by

the inventors, they have conceived and reduced to practice a software and/or hardware facility for maintaining a cardinality schema to manage indications of property or predicate cardinality on behalf of a database that is indifferent to the cardinality of the properties or predicates stored within it ("the facility"). To simplify the discussion herein, graph databases, subject-predicate-object databases, triplestores, and databases of other types that are indifferent to cardinality are referred to hereafter as "graph databases," or simply as "databases." For the same reason, both properties of entities and predicates of subjects are referred to hereafter as "properties." Also, the cardinality schema is sometimes referred to herein as a "cardinality context," or simply a "context."

[0022] In an example of the operation of the facility, an update mechanism directs an update request to the database that adds a social security number property to a particular entity, at a time when none of the entities in the database have a social security number property. When this update request is received, the facility checks whether the cardinality schema that it maintains on behalf of the database indicates a cardinality level for the social security number property. Upon determining that it does not, the facility determines that the update request includes cardinality semantics that indicate that the social security number property is a scalar property--that is, for each entity, it has at most one value. Based on this determination, the facility adds an indication to its cardinality schema that the social security number property has the scalar cardinality level. If a future update request uses cardinality semantics that indicate that the Social Security number property is a collection property–that is, it can have any number of values for a single entity– the facility responds to the update request with an error. In cases where such an error is based upon an incorrect cardinality indication in the cardinality schema, update mechanism can explicitly correct the cardinality indication in the cardinality schema. Similarly, if a future query is received from a query mechanism that contains cardinality semantics for a property that don't match the property's cardinality indication in the cardinality schema, the facility response to the query mechanism with an error.

[0023] In some embodiments, the facility automatically adds cardinality indications to its cardinality schema based upon cardinalities for properties that the facility determines from update operations involving those properties. For example, in some embodiments, the facility automatically adds a cardinality indication to the cardinality schema based upon determining the cardinality of a property from cardinality semantics included in an update operation. In some embodiments, the facility automatically adds a cardinality indication to the cardinality schema based upon determining the cardinality of a property

from the number of values associated with a property in update operation. In some embodiments, the facility permits client applications to directly manipulate the cardinality indications in the cardinality schema by explicitly creating or correcting the cardinality level indicated for particular property.

[0024] In some embodiments, where a query or update request uses cardinality semantics for a property that are inconsistent with the cardinality schema's cardinality indication for that property, the facility rejects the query or update request.

[0025] In some embodiments, the facility uses its cardinality schema to automatically complete incomplete queries to contain proper cardinality semantics for each property that is involved. For example, in some embodiments, the facility performs this automatic completion in the context of a code editor, another software development tool, or a field for entry and query to be performed immediately.

[0026] By performing some or all of the ways described above, the facility substantially increases the likelihood that the database will continue to contain cardinality-appropriate data; that update and query tools will be more accurately apprised of the cardinality of properties; that client applications will control the cardinality information available about properties to be more accurate; and/or that those forming queries for inclusion in code or immediate execution will do so with the proper cardinality semantics.

[0027] Also, by performing in some or all of the ways described above and storing, organizing, and/or accessing information relating to a database in an efficient way, the facility meaningfully reduces the hardware resources needed to store and exploit this information, including, for example: reducing the amount of storage space needed to store the information relating to the database; and reducing the number of processing cycles needed to store, retrieve, or process the information relating to the database. This allows programs making use of the facility to execute on computer systems that have less storage and processing capacity, occupy less physical space, consume less energy, produce less heat, and are less expensive to acquire and operate. Also, such a computer system can respond to user requests pertaining to information relating to a graph database with less latency, producing a better user experience and allowing users to do a particular amount of work in less time.

[0028] Figure 1 is a block diagram showing some of the components typically incorporated in at least some of the computer systems and other devices on which the facility operates. In various embodiments, these computer systems and other devices 100 can include server computer systems, desktop computer systems, laptop computer

systems, netbooks, mobile phones, personal digital assistants, televisions, cameras, automobile computers, electronic media players, etc. In some embodiments, the facility executes on one or more layers of virtual machines, which in turn execute on a physical computer system or similar device. In various embodiments, the computer systems and devices include zero or more of each of the following: a central processing unit ("CPU") 101 for executing computer programs; a computer memory 102 for storing programs and data while they are being used, including the facility and associated data, an operating system including a kernel, and device drivers; a persistent storage device 103, such as a hard drive or flash drive for persistently storing programs and data; a computer-readable media drive 104, such as a floppy, CD-ROM, or DVD drive, for reading programs and data stored on a computer-readable medium; and a network connection 105 for connecting the computer system to other computer systems to send and/or receive data, such as via the Internet or another network and its networking hardware, such as switches, routers, repeaters, electrical cables and optical fibers, light emitters and receivers, radio transmitters and receivers, and the like. While computer systems configured as described above are typically used to support the operation of the facility, those skilled in the art will appreciate that the facility may be implemented using devices of various types and configurations, and having various components.

[0029] Figure 2 is a storage diagram showing data and code stored by the facility in some embodiments. A storage device 200 – such as memory 102, persistent storage 103, or media accessed via a computer readable media drive 104 – stores an engine 210 and data 224 for a graph database or a database of another similar type. The graph database data includes substantive data 221 stored natively by the database -- such as substantive graphs, triples, or other representations – and a cardinality schema 222 stored separately from the substantive data. In some embodiments (not shown), the facility stores the cardinality schema outside the graph database data, and/or manages the cardinality schema using code other than the graph database engine.

[0030] Figure 3 is a flow diagram showing a process performed by the facility in some embodiments to process update requests. In act 301, the facility receives an update request. The update request identifies an entity in a graph database. For each of one or more properties, the request specifies adding or replacing one or more values for the property to the entity. In some embodiments, where the update request does not identify an existing entity, the request is interpreted to create an entity having the specified property values.

[0031] Table 1A below shows a sample update request received at a time when the cardinality schema and database content are both empty. Because it does not identify any existing entity in the state of the database at the time the request is received, it is interpreted to create a new entity having the single value "Compilers: Principles, Techniques and Tools" for a “title” property; the two values "Alfred V. Aho" and "Ravi Sethi" for an “author” property; and the single value "978-0321486813" for an “isbn” property. The sample update request in Table 1A also includes cardinality semantics: the brackets surrounding values of the “author” property are semantic indications that the “author” property has the collection cardinality level; the absence of brackets surrounding values of the “title” and “isbn” properties are semantic indications that these two properties have the scalar cardinality level.

```
{
  "title": "Compilers: Principles, Techniques and Tools",
  "author": [ "Alfred V. Aho", "Ravi Sethi" ],
  "isbn": "978-0321486813"
}
```

Table 1A – Request 1: POST to Content

[0032] In acts 302-310, the facility loops through each property for which the update request specifies values. In the request shown above in Table 1A, these are "title," "author," and "isbn." In act 303, if the cardinality schema already indicates the cardinality of the current property, then the facility continues in act 307, else the facility continues in act 304. In act 304, if the semantics of the received update request specify the cardinality of the current property, then the facility continues in act 305, else the facility continues in act 306. In act 305, the facility records the cardinality for the current property in the cardinality schema based upon the cardinality semantics of the request. After act 305, the facility continues in act 308.

[0033] Figure 4 is a table diagram showing logic used by the facility in some embodiments to perform act 305. A table 400 shows, in the situation where the facility receives an update request specifying one or more values of a property called “X” whose cardinality is not indicated by the cardinality schema, that the update operation always succeeds for the “X” property. The table also shows, for various combinations of the cardinality indicated for the property by semantics of the update request (rows 401 and 402) with number of values of the property (columns 412, 413, and 414), the cardinality recorded in the cardinality schema for the “X” property. For example, the entry at the

intersection of row 401 with column 412 indicates that, when an update request to post a single value to the scalar property "X" is received when the database contains no values for the property "X" and the cardinality schema contains no cardinality indication for the property "X," the update operation succeeds with respect to the "X" property, and that the facility records an indication in the cardinality schema indicating that the "X" property has the scalar cardinality level.

[0034] While Figure 4 and each of the table diagrams discussed below show a table whose contents and organization are designed to make them more comprehensible by a human reader, those skilled in the art will appreciate that actual data structures used by the facility to store this information may differ from the table shown, in that they, for example, may be organized in a different manner; may contain more or less information than shown; may be compressed, encrypted, and/or indexed; may contain a much larger number of rows than shown, etc.

[0035] Returning to Figure 3, in act 306, the facility records the cardinality for the property in the cardinality schema based instead upon the number of values of the property specified by the request. After act 306, the facility continues in act 308. In act 307, if the number of values specified for the property in the request and the semantics of the request with respect to the cardinality of the property, together with the update operation type, are consistent with the cardinality of the property indicated by the cardinality schema, then the facility continues in act 308, else the facility continues in act 309.

[0036] Figures 5 and 6 are table diagrams showing logic used by the facility in some embodiments to perform act 307. In Figure 5, a table 500 shows, in the situation where the facility receives an update request specifying one or more values of a property called "X" whose cardinality level is indicated by the cardinality schema to be scalar, whether the update request succeeds or fails in various situations. Each of these situations is a combination of the cardinality indicated for the property by semantics of the update request (rows 501 and 502) with number of values of the property (columns 512, 513, and 514). For example, the entry at the intersection of row 401 with column 513 indicates that, when an update request to post a single value to the scalar property "X" is received when the database contains one value for the property "X" and the context schema indicates that "X" is a scalar property, the update operation fails with respect to the "X" property.

[0037] In Figure 6, a table 600 shows, in the situation where the facility receives an update request specifying one or more values of a property called "X" whose cardinality

level is indicated by the cardinality schema to be collection, whether the update request succeeds or fails in various situations. Each of these situations is a combination of the cardinality indicated for the property by semantics of the update request (rows 601 and 602) with number of values of the property (columns 612, 613, and 614). In particular, all of the entries of the table indicate that the update operation always succeeds with respect to the "X" property. This is because the update operation "post" is additive, in the sense that it adds any specified property values not already possessed by the entity that is the subjection of the post operation. The post of any number of values is always proper with respect to a property that is known to be a collection.

[0038] Returning to Figure 3, in act 308, the facility performs the update operation with respect to the property – that is, it associates the specified values with the property for the entity explicitly or implicitly identified by the update request. This association is performed in a way specified by the particular update operation identified by the update request; for example, for a Post update operation, the facility adds a new value for the combination of entity and property, while for a Patch update operation, the facility replaces any existing value(s) for the combination of entity and property with the value or values specified by the update request. After act 308, the facility continues in act 310. In act 309, the facility raises an error with respect to the current property. In act 310, where additional properties remain to be processed, the facility continues in 302 the process the next property. After act 310, this process concludes. In some embodiments (not shown), the facility organizes the activities shown in Figure 3 such that cardinality validation is first performed across all of the properties included in the update request, and processing the requested update occurs only if validation succeeds for all of the included properties.

[0039] Those skilled in the art will appreciate that the acts shown in Figure 3 and in each of the flow diagrams discussed below may be altered in a variety of ways. For example, the order of the acts may be rearranged; some acts may be performed in parallel; shown acts may be omitted, or other acts may be included; a shown act may be divided into subacts, or multiple shown acts may be combined into a single act, etc.

[0040] For each of the properties in the sample update request shown in Table 1A, the facility traverses through acts 303 and 304 to act 305-- based upon the cardinality schema being empty and the request containing property cardinality semantics for all of the properties -- to record in the cardinality schema a cardinality level for each of the three properties that is based upon the request's cardinality semantics, as shown below in Table 1B. In particular, the cardinality schema reflects that (1) the author property is a collection

property, as the result of brackets surrounding the list of values specified for it by the update request, and that (2) the title and isbn properties are scalar properties, as a result of brackets not surrounding the value specified for either property by the update request.

subject	predicate	object
title	type	ScalarProperty
author	type	CollectionProperty
isbn	type	ScalarProperty

Table 1B – Schema State

[0041] For each of the properties in the sample update request shown in Table 1A, the facility further traverses from act 305 to act 308, where the facility performs the requested update operation with respect to the property. The resulting state of the database, which was empty at the time the request was received, is shown below in Table 1C. The entity “1234” that is the subject of each of the triple show in Table 1C is assigned by the database to uniquely identify the entity created in response to the request. In some embodiments, the database returns information about this created entity to facilitate future access to it by the client applications, such as the identifier "1234."

subject	predicate	object
1234	title	"Compilers: Principles, Techniques and Tools"
1234	author	"Alfred V. Aho"
1234	author	"Ravi Sethi"
1234	isbn	"978-0321486813"

Table 1C – Content State

[0042] Figure 7 is a graph diagram showing the contents of the database corresponding to Table 1C in graph form. The graph 700 has an entity node 710 created in response to the request. The entity node is connected to property value node 730 by edge 720, connoting that the “1234” entity has the value "Compilers: Principles, Techniques and Tools" for the “title” property. Edges 740 and 760 each connect the entity node to a different property value node (450 or 770) for the “author” property. Finally, edge 780 connects the entity node to property value node 790, connoting that the “1234” entity has the value "978-0321486813" for the “isbn” property. As noted above, content can generally be straightforwardly converted in either direction between subject-predicate-object database form and graph database form.

[0043] Five more sample requests that build on the one shown in Table 1A are

shown in Tables 2A, 3A, 4A, 5A, and 6A, and discussed below. Tables 2B, 3B, 4B, 5B, and 6B track the state of the cardinality schema in response to each additional sample request, while Tables 2C, 3C, 4C, 5C, and 6C track the corresponding state of the database.

[0044] Table 2A below shows a second content post request, received at a time when the cardinality schema has the state shown in Table 1B and the content has the state shown in Table 1C.

```
{
  "author": "Jeffrey D. Ullman",
  "publisher": "Addison-Wesley Publishing Company",
  "isbn": "978-0321486813"
}
```

Table 2A – Request 2: POST to Content

[0045] When the facility receives the update request shown in Table 2A, it uses the value of the isbn property specified by the request to correlate the update request with entity 1234, which has this value of the isbn property. The effect of the request shown in Table 2A on the cardinality schema is shown below in Table 2B, where an indication is added that a new publisher property has the scalar cardinality schema, as a result of the fact that the value specified for the new publisher property is not surrounded by brackets in the update request, and thus has cardinality semantics indicating the scalar cardinality level.

subject	**predicate**	**object**
title	type	ScalarProperty
author	type	CollectionProperty
isbn	type	ScalarProperty
publisher	type	ScalarProperty

Table 2B – Schema State

[0046] The update request shown in Table 2A causes the facility to update the content as shown below in Table 2C. In particular, a third value of the author property is added, as well as a value for the new publisher property.

subject	**predicate**	**object**
1234	title	"Compilers: Principles, Techniques and Tools"
1234	author	"Alfred V. Aho"

subject	predicate	object
1234	author	"Ravi Sethi"
1234	isbn	"978-0321486813"
1234	author	"Jeffrey D. Ullman"
1234	publisher	"Addison-Wesley Publishing Company"

Table 2C – Content State

[0047] Table 3A below shows a third content post request, received at a time when the cardinality schema has the state shown in Table 2B and the content has the state shown in Table 2C. The content publisher request seeks to add the value "Monica S. Lam" to the values of the author property.

```
{
  "author": "Monica S. Lam",
  "isbn": "978-0321486813"
}
```

Table 3A – Request 3: POST to Content

[0048] This update has no effect on the cardinality schema; thus, Table 3B below is unchanged from Table 2B above.

subject	predicate	object
title	type	ScalarProperty
author	type	CollectionProperty
isbn	type	ScalarProperty
publisher	type	ScalarProperty

Table 3B – Schema State

[0049] The update is consistent with the collection cardinality level indicated for the author property by the cardinality schema (Properties having the collection cardinality level can have any number of values for a single entity.), and thus is applied to the content state. Table 3C below shows the addition of the new value for the author property.

subject	predicate	object
1234	title	"Compilers: Principles, Techniques and Tools"
1234	author	"Alfred V. Aho"
1234	author	"Ravi Sethi"
1234	isbn	"978-0321486813"
1234	author	"Jeffrey D. Ullman"

1234	publisher	"Addison-Wesley Publishing Company"
1234	author	"Monica S. Lam"

Table 3C – Content State

[0050] Table 4A below shows a fourth content post request, received at a time when the cardinality schema has the state shown in Table 3B and the content has the state shown in Table 3C.

```
{
  "publisher": "Bellevue Publishing Company",
  "isbn": "978-0321486813"
}
```

Table 4A – Request 4: POST to Content

[0051] When this update request is received, the facility rejects it, as it constitutes a request to add a second value to the title property, which is indicated by the cardinality schema shown in Table 3C above to have the scalar cardinality level. (Properties having the scalar cardinality level can have only one value for a single entity.) This rejected request has no effect on the cardinality schema or the content, as shown below in Tables 4B and 4C, respectively.

subject	**predicate**	**object**
title	type	ScalarProperty
author	type	CollectionProperty
isbn	type	ScalarProperty
publisher	type	ScalarProperty

Table 4B – Schema State

subject	**predicate**	**object**
1234	title	"Compilers: Principles, Techniques and Tools"
1234	author	"Alfred V. Aho"
1234	author	"Ravi Sethi"
1234	isbn	"978-0321486813"
1234	author	"Jeffrey D. Ullman"
1234	publisher	"Addison-Wesley Publishing Company"
1234	author	"Monica S. Lam"

Table 4C – Content State

[0052] Where a client application discovers by the rejection of an update request on

the basis of a cardinality level mismatch that the cardinality schema is out of sync with the cardinality level intended by the client application for a particular property, the client application can explicitly alter the cardinality indication for that property in the cardinality schema. Figure 8 is a flow diagram showing a process performed by the facility in some embodiments to alter the cardinality schema in response to an explicit request from a client application. In act 801, the facility receives a request to set a cardinality indication for a specified property in the cardinality schema to a particular cardinality level, such as scalar or collection. In act 802, the facility sets the cardinality indication for the property specified by the request to the cardinality level property specified by the request. In some embodiments (not shown), the facility performs one or more validation tests before setting the cardinality indication in act 802. For example, in some embodiments, the facility determines whether, for some or all of the entities in the database, the number of values for the specified attribute is compatible with the cardinality level specified for this attribute by the request. In various embodiments, when such tests identify inconsistencies, the facility responds by failing the update operation; warning the client application; logging the inconsistencies; etc. After act 802, this process concludes.

[0053] Table 5A below shows a fifth update request, this one a request to patch the cardinality schema to change the cardinality indication for the publisher property from scalar to collection. It is received at a time when the cardinality schema has the state shown in Table 4B and the content has the state shown in Table 4C.

```
{
  "id": "publisher",
  "type": "CollectionProperty"
}
```

Table 5A – Request 5: PATCH to Schema

[0054] In response to the request, the facility changes the cardinality indication for the publisher property in the context schema from scalar to collection, as shown below in Table 5B .

subject	**predicate**	**object**
title	type	ScalarProperty
author	type	CollectionProperty
isbn	type	ScalarProperty
publisher	type	CollectionProperty

Table 5B – Schema State

[0055] The facility does not alter the content state in response to this request; thus, Table 5C below matches Table 4C above.

subject	predicate	object
1234	title	"Compilers: Principles, Techniques and Tools"
1234	author	"Alfred V. Aho"
1234	author	"Ravi Sethi"
1234	isbn	"978-0321486813"
1234	author	"Jeffrey D. Ullman"
1234	publisher	"Addison-Wesley Publishing Company"
1234	author	"Monica S. Lam"

Table 5C – Content State

[0056] Table 6A below shows a sixth update request, this one a repetition of the fourth request seeking to add an additional value to the publisher schema that is shown in Table 4A above. It is received at a time when the cardinality schema has the state shown in Table 5B and the content has the state shown in Table 5C.

```
{
  "publisher": "Bellevue Publishing Company",
  "isbn": "978-0321486813"
}
```

Table 6A – Request 6: POST to Content

[0057] The request does not alter the cardinality schema; thus, Table 6B below matches Table 5B above.

subject	predicate	object
title	type	ScalarProperty
author	type	CollectionProperty
isbn	type	ScalarProperty
publisher	type	CollectionProperty

Table 6B – Schema State

[0058] Because the cardinality indication for the publisher property has been changed to collection, which can accommodate multiple values, the update is successful, adding the value "Bellevue Publishing Company" for the publisher property to the existing value of "Addison-Wesley Publishing Company" as shown below in Table 6C.

subject	predicate	object
1234	title	"Compilers: Principles, Techniques and Tools"
1234	author	"Alfred V. Aho"
1234	author	"Ravi Sethi"
1234	isbn	"978-0321486813"
1234	author	"Jeffrey D. Ullman"
1234	publisher	"Addison-Wesley Publishing Company"
1234	author	"Monica S. Lam"
1234	publisher	"Bellevue Publishing Company"

Table 6C – Content State

[0059] Figure 9 is a flow diagram showing a process performed by the facility in some embodiments to respond to a query against the database. In act 901, the facility receives a query seeking to identify entities that in a graph database or database of another type that have property values matching value specified for one or more properties by the query. In acts 902-908, the facility loops through each property for which the query specifies one or more values. In act 903, if semantics of the query specify a cardinality level for the property, then the facility continues in act 904, else the facility continues in act 907. In act 904, if the cardinality schema indicates a cardinality level for the property, then the facility continues in act 905, else the facility continues in act 907. In act 905, if the cardinality level of the property specified by the query semantics is consistent with the cardinality of the property indicated by the cardinality schema, then the facility continues in act 907, else the facility continues in act 906 to return an error in conclude the process. In act 907, the facility adds to a result set for the current property each entity in the database that has values for the current property that match the value specified for this property by the query. In act 908, if additional properties remain to be processed, then the facility continues in act 902 to address the next property, else the facility continues in act 909. In act 909, the facility returns a result that includes each entity that is present in all of the result sets established in act 907. After act 909, this process concludes.

[0060] Figures 10-12 are table diagrams showing logic used by the facility in some embodiments to perform acts 905 and 907. In Figure 10, a table 1000 shows, in the situation where the facility receives a query specifying one or more values of a property called “X” whose cardinality level is not indicated by the cardinality schema, whether the query is in error, and whether it matches an entity having a particular combination of one

or more values for the property. In particular, the table specifies such a result for each a combination of the cardinality indicated for the property by semantics of the query (rows 1001 and 1002) with number of values and particular values of the property associated with a particular entity (columns 1012-1016). For example, the entry at the intersection of row 1001 with column 1014 indicates that, when a query specifying a single value 42 for the scalar property "X" is received when the database contains the multiple values 42 and 64 for the property "X" for the current entity and the context schema indicates no cardinality level for the property "X," the query fails for the entity with respect to the "X" property, as the actual number of values for the property and entity is mismatched with the cardinality semantics of the query.

[0061] In Figure 11, a table 1100 shows, in the situation where the facility receives a query specifying one or more values of a property called "X" whose cardinality level is indicated to be scalar by the cardinality schema, whether the query is in error, and whether it matches an entity having a particular combination of one or more values for the property. In particular, the table specifies such a result for each a combination of the cardinality indicated for the property by semantics of the query (rows 1101 and 1102) with number of values and particular values of the property associated with a particular entity (columns 1112-1116). For example, the entry at the intersection of row 1102 with column 1112 indicates that, when a query specifying a single value 42 for the collection property "X" is received when the database contains the multiple no values for the property "X" for the current entity, and the context schema indicates the scalar cardinality level for the property "X," the query fails for the entity with respect to the "X" property, as the cardinality schema indication for the property is mismatched with the cardinality semantics of the query.

[0062] In Figure 12, a table 1200 shows, in the situation where the facility receives a query specifying one or more values of a property called "X" whose cardinality level is indicated to be collection by the cardinality schema, whether the query is in error, and whether it matches an entity having a particular combination of one or more values for the property. In particular, the table specifies such a result for each a combination of the cardinality indicated for the property by semantics of the query (rows 1201 and 1202) with number of values and particular values of the property associated with a particular entity (columns 1212-1216). For example, the entry at the intersection of row 1201 with column 1212 indicates that, when a query specifying a single value 42 for the scalar property "X" is received when the database contains the multiple no values for the property "X" for the

current entity, and the context schema indicates the collection cardinality level for the property "X," the query fails for the entity with respect to the "X" property, as the cardinality schema indication for the property is mismatched with the cardinality semantics of the query.

[0063] Figure 13 is a flow diagram showing a process performed by the facility in some embodiments to automatically complete incomplete queries using proper cardinality semantics for the involved properties. In act 1301, the facility monitors for the entry of a partial entity query that specifies values for one or more properties. In various embodiments, the facility performs this monitoring in various data entry contexts such as in a code editor, in a software development tool another type, in a real-time query field, etc.

[0064] Figure 14 is a display diagram showing a sample display depicting the entry of an incomplete query. The display 1400 includes a code editor window 1410 into which the user has entered incomplete query 1411. The incomplete query includes text "$filter=author" indicating that the completed query is intended to select entities on the basis of their values of an author property.

[0065] Returning to Figure 13, in act 1302, among the properties for which the partial query identified in act 1301 specifies values, the facility identifies any of these properties for which the cardinality schema indicates the property's cardinality. In act 1303, the facility uses the identified partial query to construct, on a predictive basis, a full query in which cardinality semantics for the properties identified in act 1302 are consistent with the cardinality levels indicated for these properties by the cardinality schema. In act 1304, the facility proposes the full query constructed in act 1303 as a substitute for the partial query entered by the user. After act 1304, this process concludes.

[0066] Figure 15 is a display diagram showing a sample display presented by the facility in some embodiments to propose a constructed predictive full query as a substitute for an entered partial query. It can be seen that the facility has replaced partial query 1411 with completed query 1511. In particular, the facility has replaced the text "$filter=author" with the text "filter=author/any(a: a eq 'Alfred V. Aho')". This replacement text includes a portion "/any(a:a . . .)" that constitutes the cardinality semantics for the collection cardinality level indicated by the cardinality schema for the author attribute. (See Table 6B.) In some embodiments (not shown), rather than including the value for the author property to be searched for 'Alfred V. Aho' in the full query, the facility includes a blank that prompts the user to type this value. In some embodiments

(not shown), the facility waits to generate a completed query until the partial query includes a value for each included property; for example, the facility would wait until the partial query said "http://fabric.com/sets/<set-id>/content/<node-id>?$expand=books($filter=author eq 'Alfred V. Aho')"; the facility would then correct these scalar cardinality semantics for the author property to the collection cardinality semantics in completed query 1511.

[0067] As an additional example, the facility in some embodiments would complete a partial query directed to the scalar isbn property as "http://fabric.com/sets/<set-id>/content/<node-id>?$expand=books($filter=isbn eq '978-0321486813')", which contains the correct scalar cardinality semantics for the isbn property.

[0068] In some embodiments, the facility provides a method in a computing system, comprising: receiving a query against a distinguished property contained by a graph database, the query reflecting a level of cardinality for the distinguished property; determining that the level of cardinality for the distinguished property reflected by the query is inconsistent with a level of cardinality of the distinguished property stored distinctly from the graph database in a cardinality schema for the graph database; and in response to determining that the level of cardinality for the distinguished property reflected by the query is inconsistent with the level of cardinality of the distinguished field stored in the cardinality schema, responding to the received query with an error indication.

[0069] In some embodiments, the facility provides one or more instances of computer-readable media collectively having contents configured to cause a computing system to perform a method, the method comprising: receiving a query against a distinguished property contained by a graph database, the query reflecting a level of cardinality for the distinguished property; determining that the level of cardinality for the distinguished property reflected by the query is inconsistent with a level of cardinality of the distinguished property stored distinctly from the graph database in a cardinality schema for the graph database; and in response to determining that the level of cardinality for the distinguished property reflected by the query is inconsistent with the level of cardinality of the distinguished field stored in the cardinality schema, responding to the received query with an error indication.

[0070] In some embodiments, the facility provides one or more instances of computer-readable media collectively having contents configured to cause a computing system to perform a method, the method comprising: receiving a first update request for a graph database, the first update request identifying an entity, a property of the entity, and

at least one value to be associated with the combination of the entity and the property, the semantics of the update request reflecting a first cardinality level for the property; determining that the first cardinality level for the property reflected by the semantics of the first update request does not match a second cardinality level stored for the property in a cardinality schema; and in response to the determining, omitting to associate the at least one value with the combination of the entity and the property in accordance with the first update request.

[0071] In some embodiments, the facility provides a method in a computing system, comprising: receiving a first update request for a graph database, the first update request identifying an entity, a property of the entity, and at least one value to be associated with the combination of the entity and the property, the semantics of the update request reflecting a first cardinality level for the property; determining that the first cardinality level for the property reflected by the semantics of the first update request does not match a second cardinality level stored for the property in a cardinality schema; and in response to the determining, omitting to associate the at least one value with the combination of the entity and the property in accordance with the first update request.

[0072] In some embodiments, the facility provides one or more instances of computer-readable media collectively storing a cardinality context repository data structure for a triplestore, the data structure comprising: for each of one or more predicates that, for a subject in the triplestore, can relate at least one object to the subject: an indication of whether the predicate can relate more than one object to a single subject, such that the contents of the data store are usable to reject requests against the triplestore for a predicate that are inconsistent with the data structure's indication for the predicate.

[0073] In some embodiments, the facility provides a method in a computing system, comprising: receiving input specifying a partial query against a distinguished property in a graph database; accessing a cardinality schema to determine a cardinality of the distinguished property in the graph database; generating from the partial query specified by the user input a completed query whose semantics reflect the cardinality determined for the distinguished property; and presenting the generated completed query as a predictive completion of the partial query.

[0074] In some embodiments, the facility provides one or more instances of computer-readable media collectively having contents configured to cause a computing system to perform a method, the method comprising: receiving input specifying a partial query against a distinguished property in a graph database; accessing a cardinality schema

to determine a cardinality of the distinguished property in the graph database; generating from the partial query specified by the user input a completed query whose semantics reflect the cardinality determined for the distinguished property; and presenting the generated completed query as a predictive completion of the partial query.

[0075] It will be appreciated by those skilled in the art that the above-described facility may be straightforwardly adapted or extended in various ways. While the foregoing description makes reference to particular embodiments, the scope of the invention is defined solely by the claims that follow and the elements recited therein.

CLAIMS

1. A method in a computing system, comprising:

receiving a query against a distinguished attribute contained by a graph database, the query reflecting a level of cardinality for the distinguished attribute;

determining that the level of cardinality for the distinguished attribute reflected by the query is inconsistent with a level of cardinality of the distinguished attribute stored distinctly from the graph database in a cardinality schema for the graph database; and

in response to determining that the level of cardinality for the distinguished attribute reflected by the query is inconsistent with the level of cardinality of the distinguished field stored in the cardinality schema, responding to the received query with an error indication.

2. The method of claim 1, further comprising:

in response to determining that the level of cardinality for the distinguished attribute reflected by the query is inconsistent with the level of cardinality of the distinguished field stored in the cardinality schema, omitting to assess whether the query against the distinguished attribute is satisfied for any entity in the graph database.

3. One or more instances of computer-readable media collectively having contents configured to cause a computing system to perform a method, the method comprising:

receiving a first update request for a graph database, the first update request identifying an entity, an attribute of the entity, and at least one value to be associated with the combination of the entity and the attribute, the semantics of the update request reflecting a first cardinality level for the attribute;

determining that the first cardinality level for the attribute reflected by the semantics of the first update request does not match a second cardinality level stored for the attribute in a cardinality schema; and

in response to the determining, omitting to associate the at least one value with the combination of the entity and the attribute in accordance with the first update request; and

responding to the update request with an error.

4. The instances of computer-readable medium of claim 3, further comprising:

after the omitting:

receiving a second update request for the cardinality schema, the second update request specifying the first cardinality level for the attribute; and

in response to receiving the second update request, altering the cardinality

schema to store the first cardinality level for the attribute.

5. The instances of computer-readable medium of claim 4, further comprising:

after the altering:

receiving a third update request for the graph database, the third update request identifying the entity, the attribute of the entity, and at least one value to be associated with the combination of the entity and the attribute, the semantics of the update request reflecting a first cardinality level for the attribute;

determining that the first cardinality level for the attribute reflected by the semantics of the first update request matches the first cardinality level stored for the attribute in a cardinality schema; and

in response to determining the matching, associate the at least one value with the combination of the entity and the attribute in accordance with the third update request.

6. One or more instances of computer-readable media collectively storing a cardinality context repository data structure for a triplestore, the data structure comprising:

for each of one or more predicates that, for a subject in the triplestore, can relate at least one object to the subject:

an indication of whether the predicate can relate more than one object to a single subject,

such that the contents of the data store are usable to reject requests against the triplestore for a predicate that are inconsistent with the data structure's indication for the predicate.

7. The instances of computer-readable medium of claim 6 wherein an indication in the data structure for a distinguished predicate was derived from a distinguished request to use the distinguished predicate to relate an object to a subject in the triplestore.

8. The instances of computer-readable medium of claim 7 wherein the indication in the data structure for the distinguished predicate was derived from semantics of the distinguished request or derived from a pair of brackets enclosing one or more objects in the distinguished request.

9. The instances of computer-readable medium of claim 6 wherein an indication in the data structure for a distinguished predicate was inferred from a distinguished request to use the distinguished predicate to relate a distinguished number of objects to a single subject.

10. A method in a computing system, comprising:

receiving input specifying a partial query against a distinguished attribute in a graph database;

accessing a cardinality schema to determine a cardinality of the distinguished attribute in the graph database;

generating from the partial query specified by the user input a completed query whose semantics reflect the cardinality determined for the distinguished attribute; and

presenting the generated completed query as a predictive completion of the partial query.

11. The method of claim 10 wherein the input received input is directed to a code editor, a software development tool other than a code editor, or a real-time query field.

12. The instances of computer-readable medium of claim 9 wherein the indication for the distinguished predicate is an indication that the predicate can relate more than one object to a single subject as the result of the distinguished request being a request to use the distinguished predicate to relate two or more objects to a single subject, or the indication for the distinguished predicate is an indication that the predicate cannot relate more than one object to a single subject as the result of the distinguished request being a request to use the distinguished predicate to relate only one object to a single subject.

13. The instances of computer-readable medium of claim 6 wherein an indication in the data structure for a distinguished predicate was derived from a distinguished request limited to specifying whether the distinguished predicate can relate more than one object to a single subject.

14. The method of claim 1 wherein:

the level of cardinality for the distinguished attribute reflected by the query is scalar and the level of cardinality of the distinguished field stored in the cardinality schema is collection, or

the level of cardinality for the distinguished attribute reflected by the query is schema and the level of cardinality of the distinguished field stored in the cardinality schema is scalar.

15. The instances of computer-readable medium of claim 3 wherein the determining determines that the first cardinality level for the attribute reflected by the semantics of the first update request, together with the number of values already associated with the combination of the entity and the attribute, does not match a second cardinality level stored for the attribute in a cardinality schema.

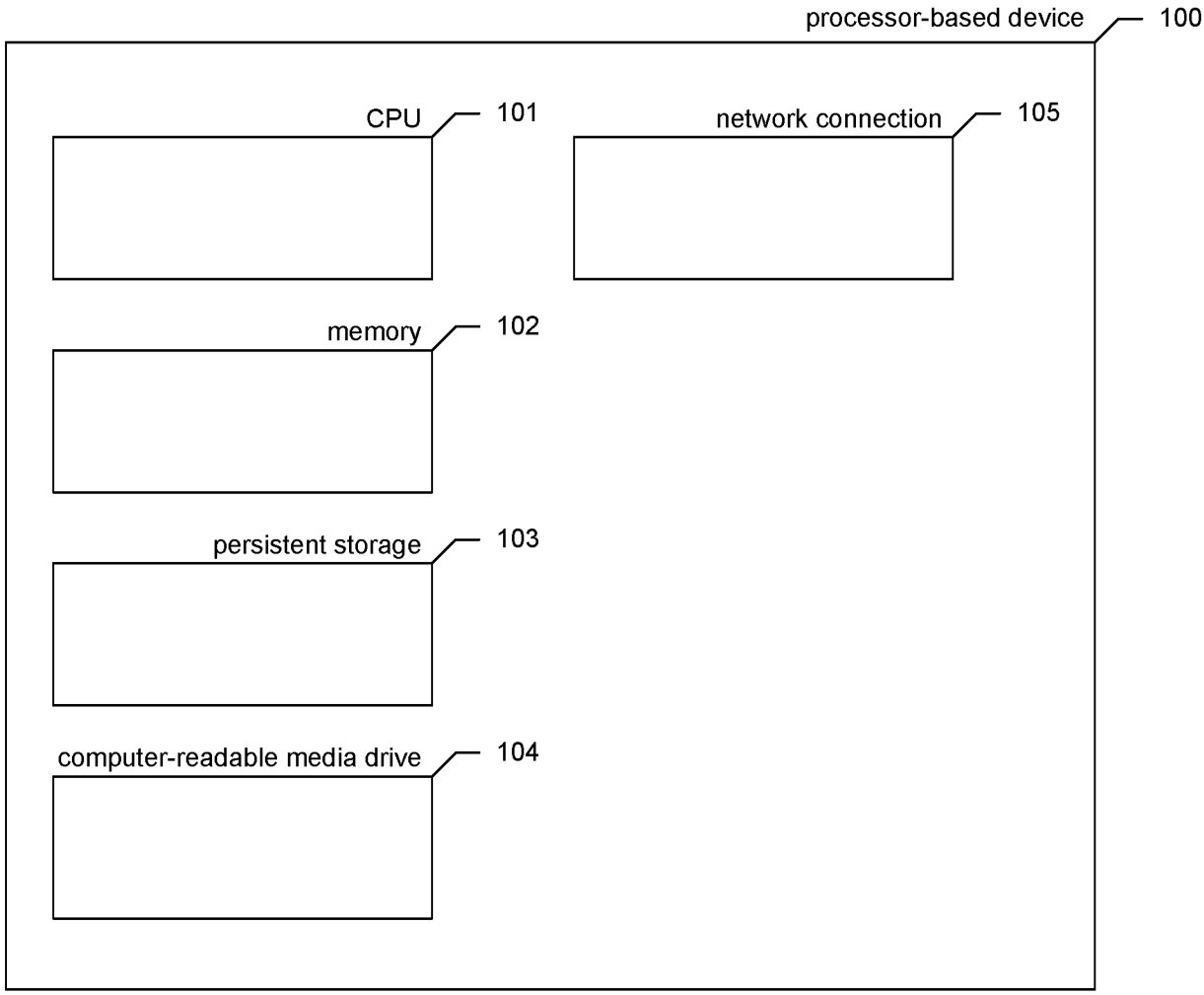
processor-based device
100
CPU
101
network connection
105
memory
102
persistent storage
103
computer-readable media drive
104

FIG. 1

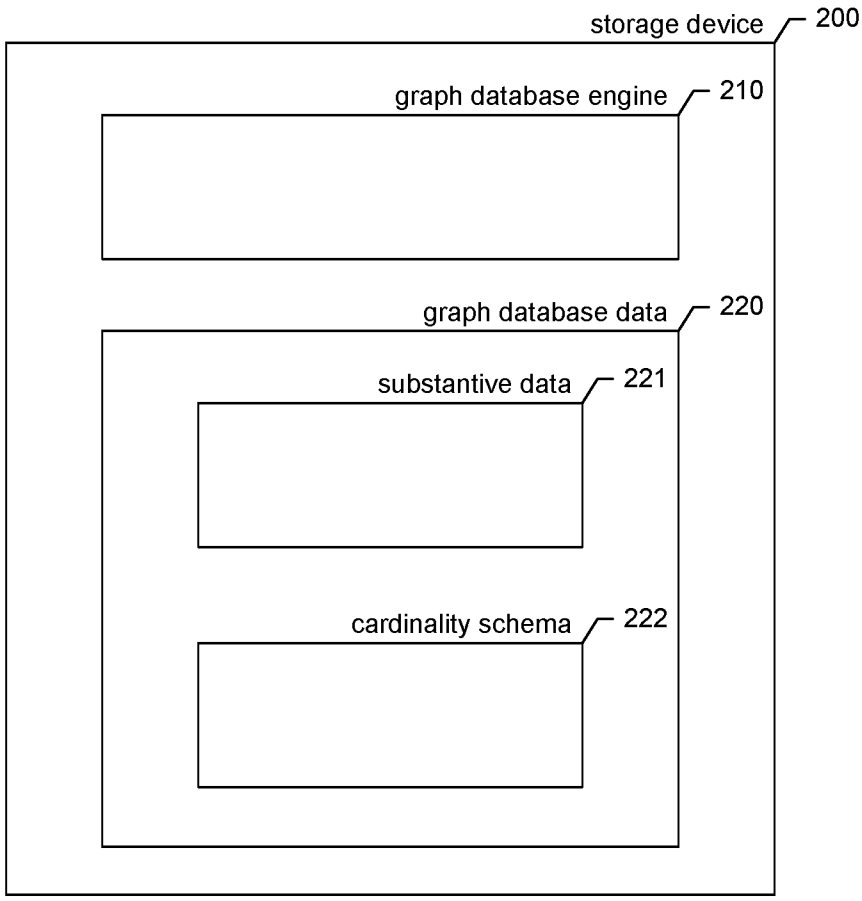
storage device
200
graph database engine
210
graph database data
220
substantive data
221
cardinality schema
222

FIG. 2

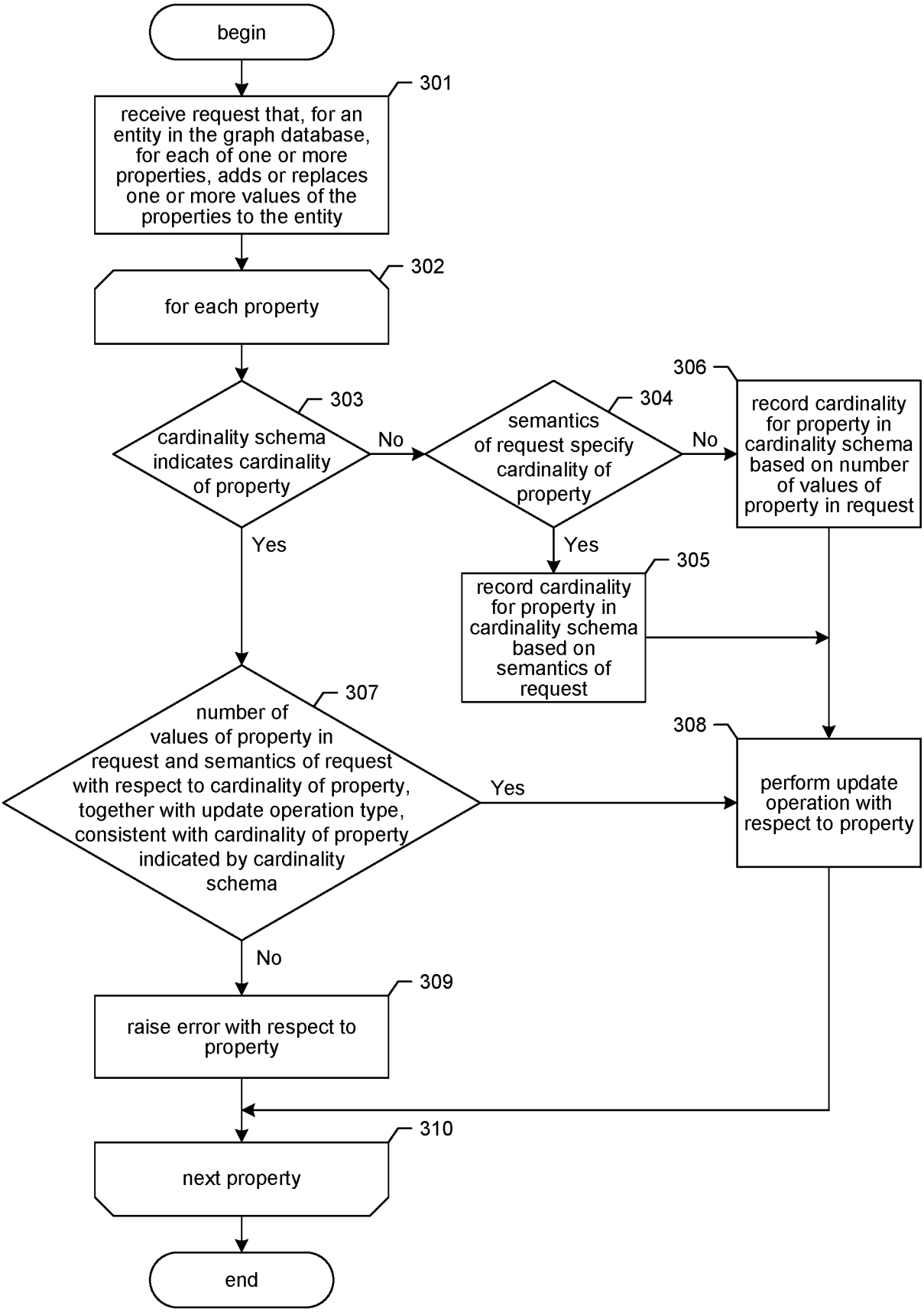
begin
301
receive request that, for an entity in the graph database, for each of one or more properties, adds or replaces one or more values of the properties to the entity
302
for each property
303
cardinality schema indicates cardinality of property
No
304
semantics of request specify cardinality of property
No
306
record cardinality for property in cardinality schema based on number of values of property in request
Yes
Yes
305
record cardinality for property in cardinality schema based on semantics of request
307
number of values of property in request and semantics of request with respect to cardinality of property, together with update operation type, consistent with cardinality of property indicated by cardinality schema
Yes
308
perform update operation with respect to property
No
309
raise error with respect to property
310
next property
end

FIG. 3

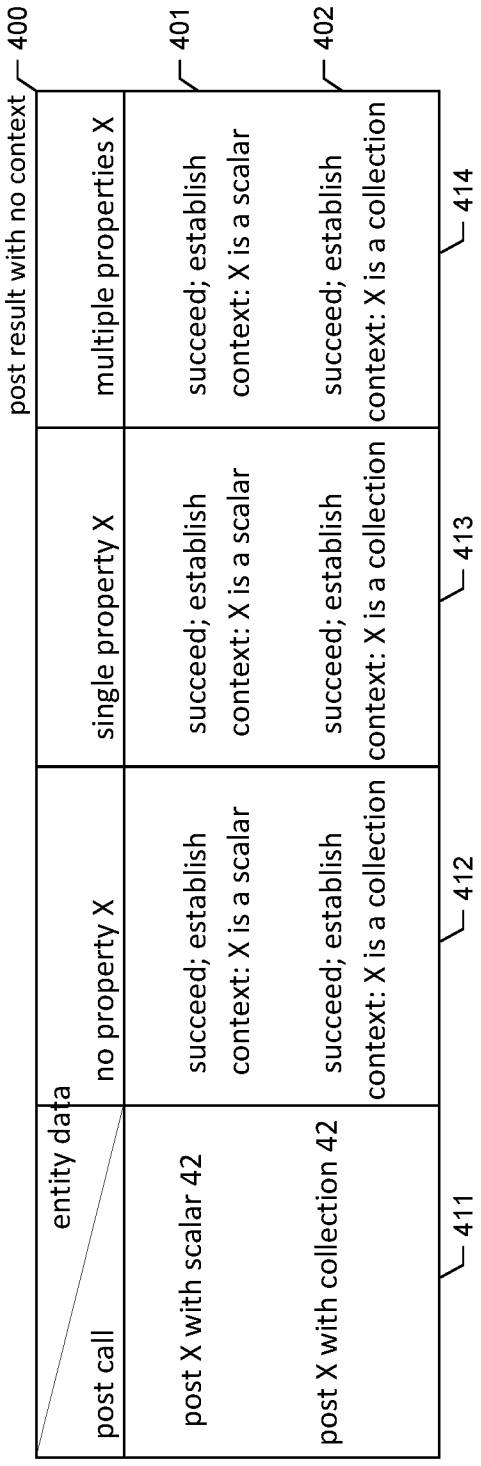
post result with no context
400
entity data
post call
no property X
single property X
multiple properties X
401
402
post X with scalar 42
succeed; establish context: X is a scalar
succeed; establish context: X is a scalar
succeed; establish context: X is a scalar
post X with collection 42
succeed; establish context: X is a collection
succeed; establish context: X is a collection
succeed; establish context: X is a collection
411
412
413
414

FIG. 4

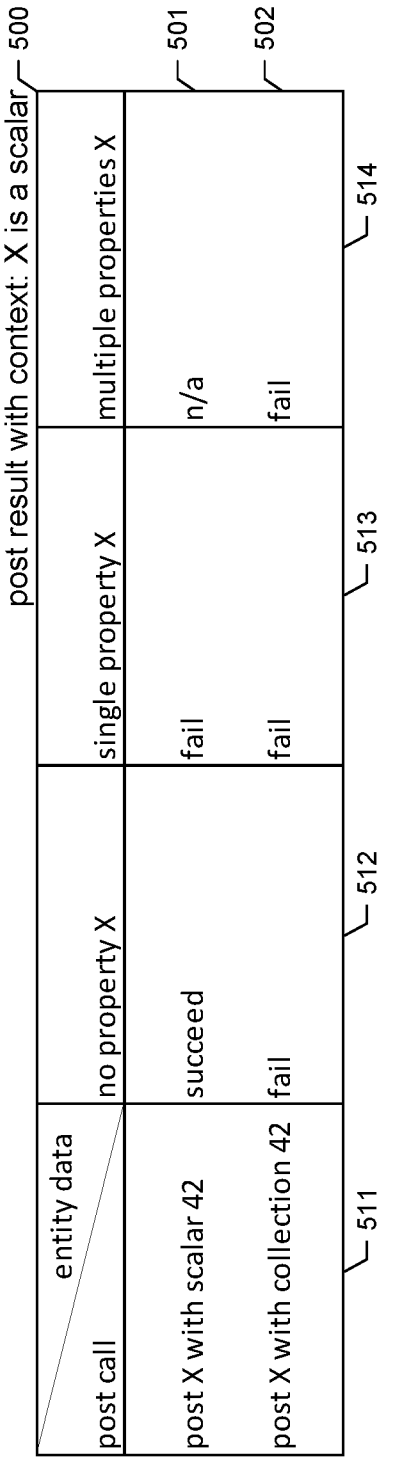
post result with context: X is a scalar
500
entity data
post call
no property X
single property X
multiple properties X
501
502
post X with scalar 42
succeed
fail
n/a
post X with collection 42
fail
fail
fail
511
512
513
514

FIG. 5

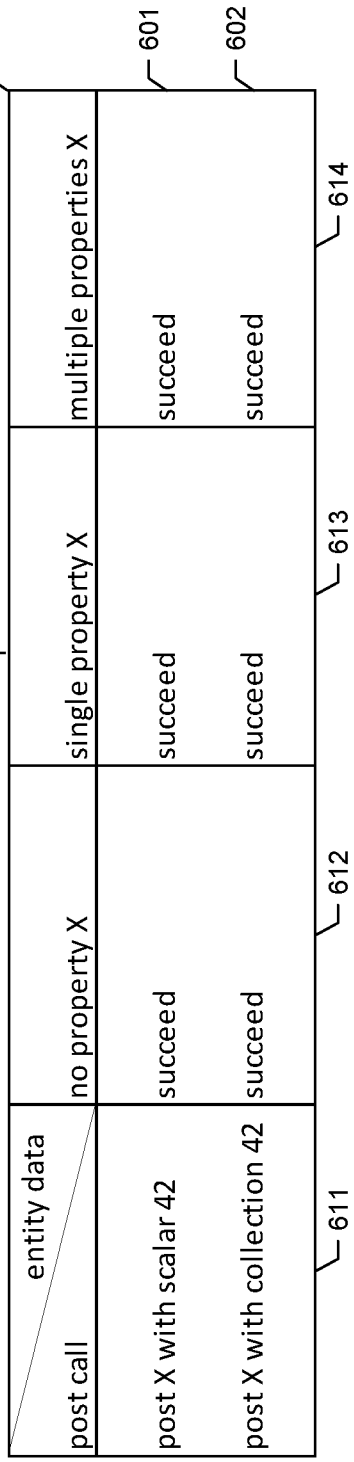
post result with context: X is a collection
600
601
602
entity data
post call
no property X
single property X
multiple properties X
post X with scalar 42
succeed
succeed
succeed
post X with collection 42
succeed
succeed
succeed
611
612
613
614

FIG. 6

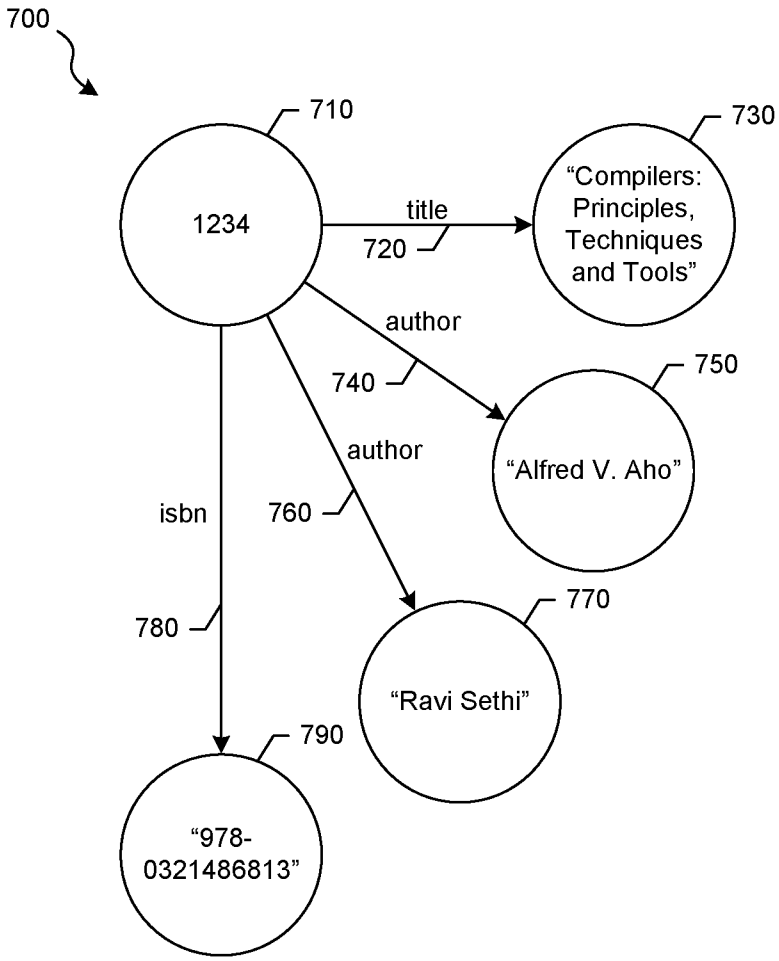
700
710
1234
730
"Compilers: Principles, Techniques and Tools"
title
720
author
740
750
"Alfred V. Aho"
author
760
isbn
780
770
"Ravi Sethi"
790
"978-0321486813"

FIG. 7

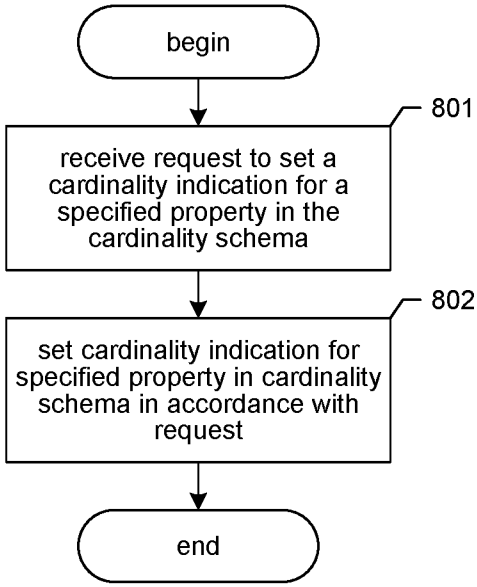
begin
801
receive request to set a cardinality indication for a specified property in the cardinality schema
802
set cardinality indication for specified property in cardinality schema in accordance with request
end

FIG. 8

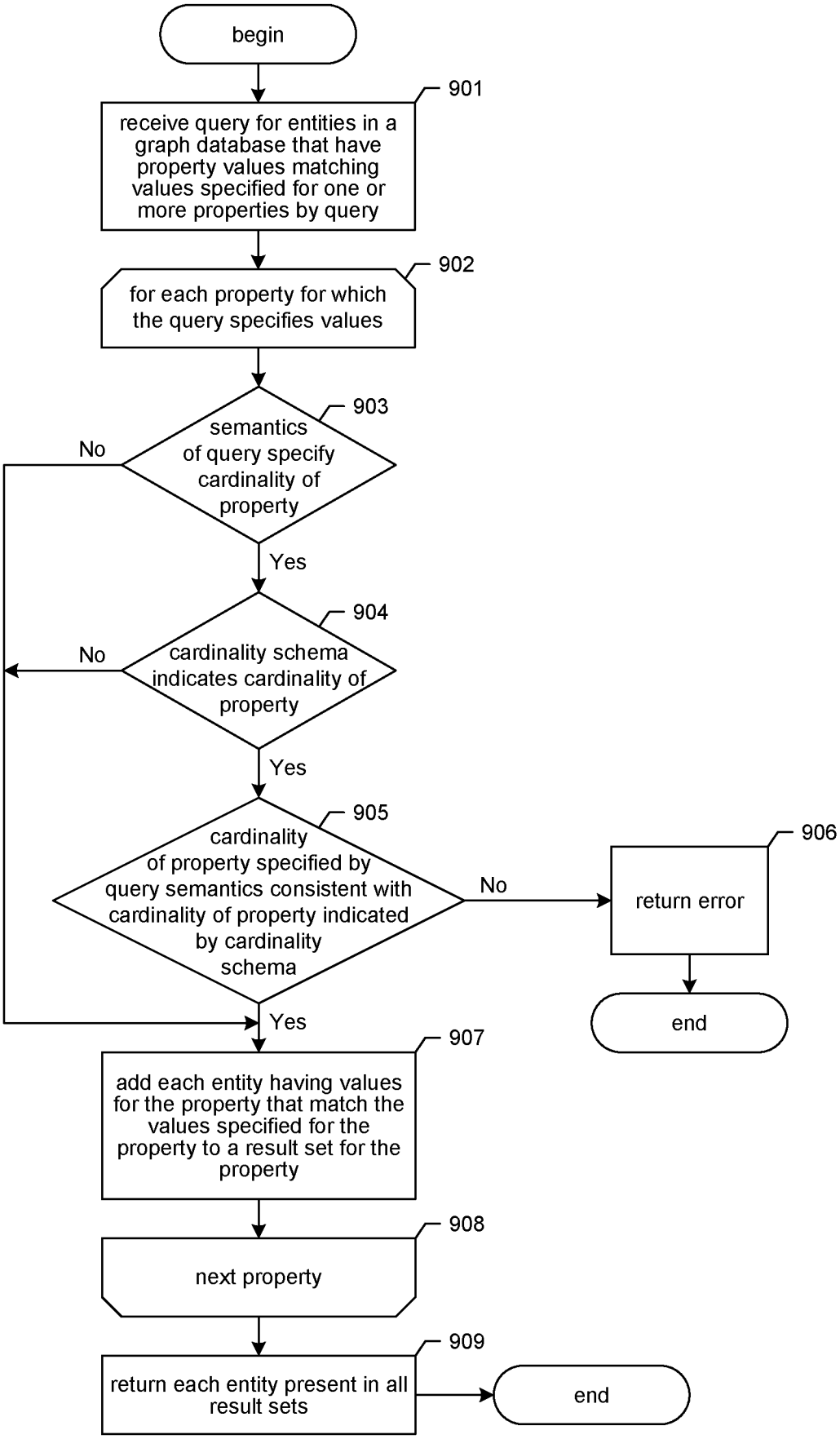
begin
901
receive query for entities in a graph database that have property values matching values specified for one or more properties by query
902
for each property for which the query specifies values
903
semantics of query specify cardinality of property
No
Yes
904
cardinality schema indicates cardinality of property
No
Yes
905
cardinality of property specified by query semantics consistent with cardinality of property indicated by cardinality schema
No
906
return error
end
Yes
907
add each entity having values for the property that match the values specified for the property to a result set for the property
908
next property
909
return each entity present in all result sets
end

FIG. 9

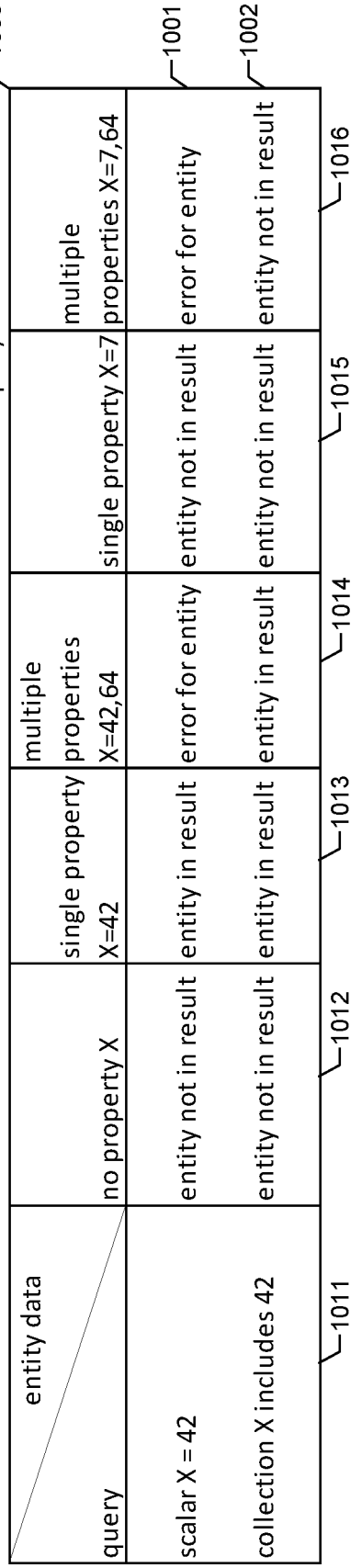
query result with no context
1000
entity data
query
no property X
single property X=42
multiple properties X=42,64
single property X=7
multiple properties X=7,64
1001
1002
scalar X = 42
entity not in result
entity in result
error for entity
entity not in result
error for entity
collection X includes 42
entity not in result
entity in result
entity in result
entity not in result
entity not in result
1011
1012
1013
1014
1015
1016

FIG. 10

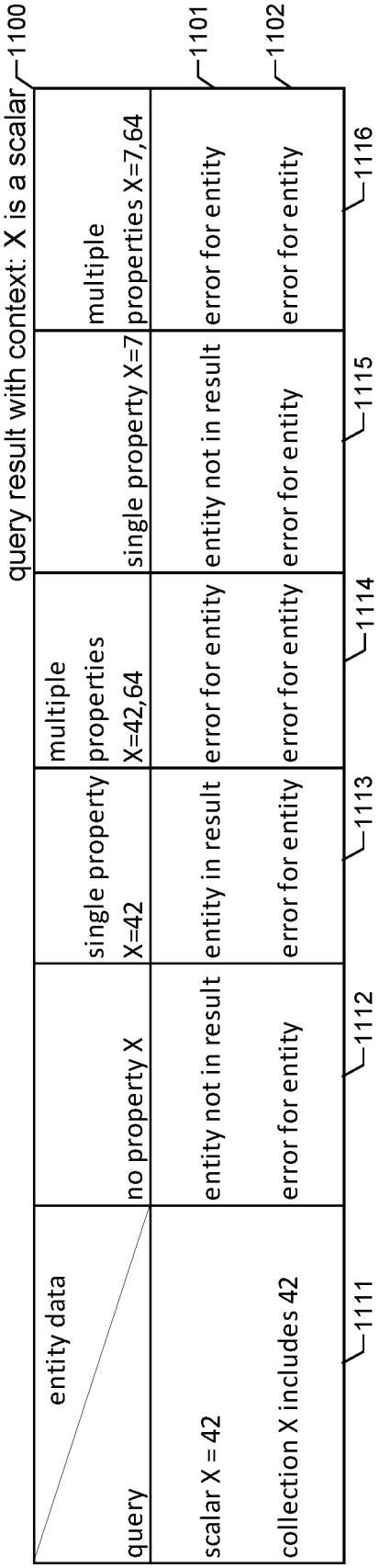
query result with context: X is a scalar
1100
entity data
query
no property X
single property X=42
multiple properties X=42,64
single property X=7
multiple properties X=7,64
1101
1102
scalar X = 42
entity not in result
entity in result
error for entity
entity not in result
error for entity
collection X includes 42
error for entity
error for entity
error for entity
error for entity
error for entity
1111
1112
1113
1114
1115
1116

FIG. 11

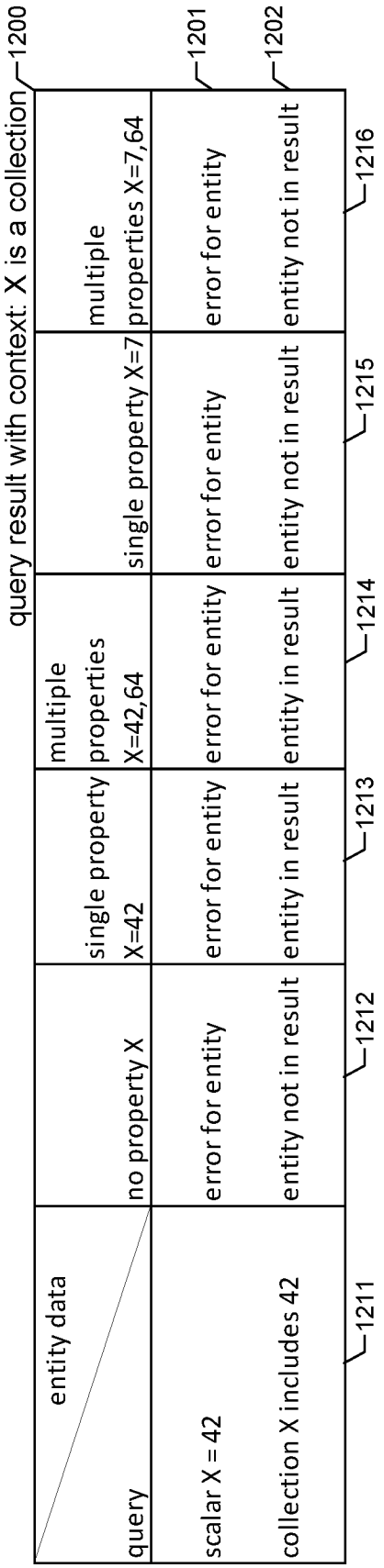
query result with context: X is a collection
1200
1201
1202
entity data
query
no property X
single property X=42
multiple properties X=42,64
single property X=7
multiple properties X=7,64
scalar X = 42
collection X includes 42
error for entity
entity not in result
error for entity
entity in result
error for entity
entity in result
error for entity
entity not in result
1211
1212
1213
1214
1215
1216

FIG. 12

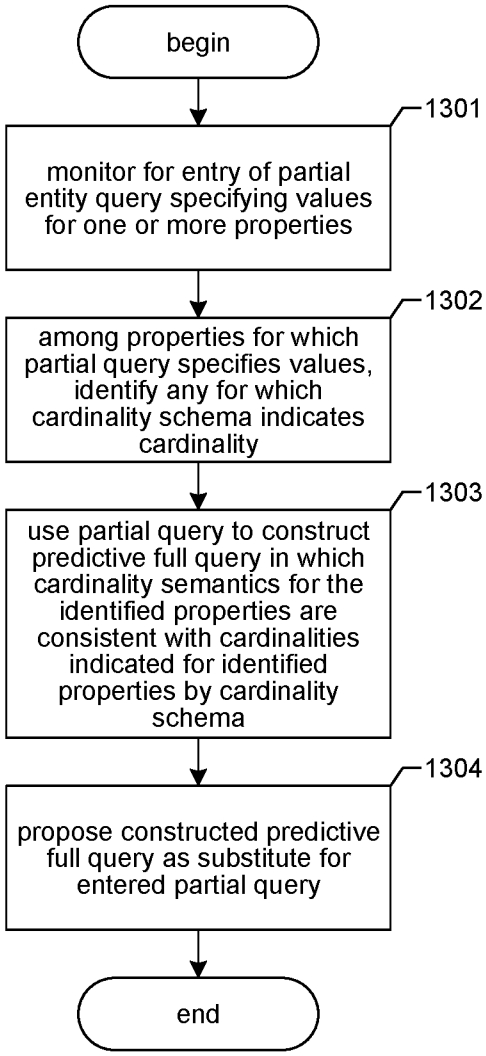
begin
1301
monitor for entry of partial entity query specifying values for one or more properties
1302
among properties for which partial query specifies values, identify any for which cardinality schema indicates cardinality
1303
use partial query to construct predictive full query in which cardinality semantics for the identified properties are consistent with cardinalities indicated for identified properties by cardinality schema
1304
propose constructed predictive full query as substitute for entered partial query
end

FIG. 13

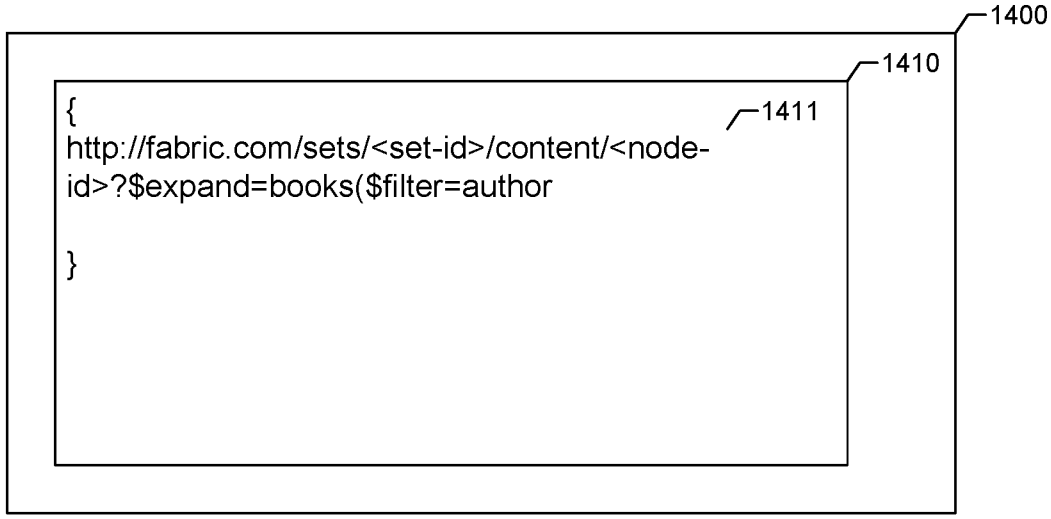
1400
1410
{
1411
http://fabric.com/sets/<set-id>/content/<node-id>?$expand=books($filter=author
}

FIG. 14

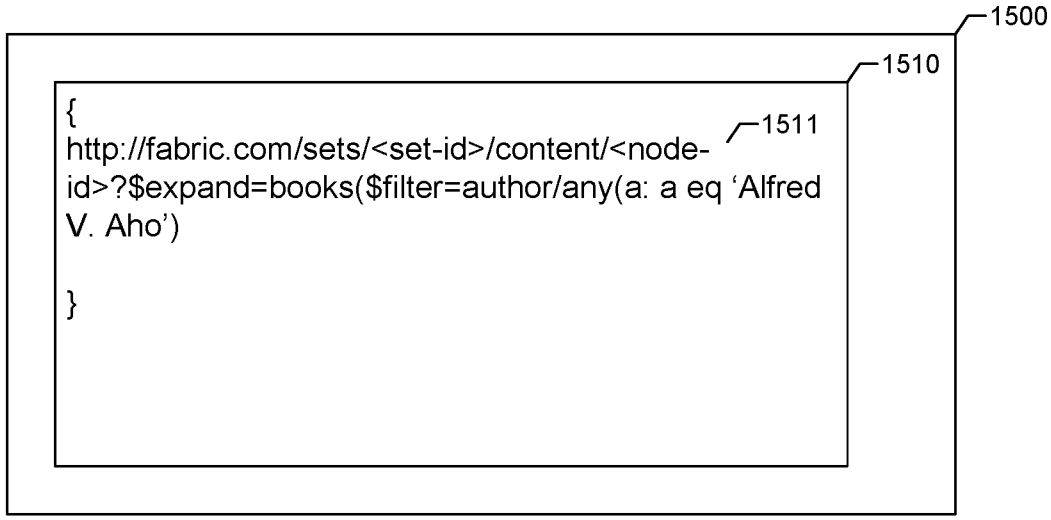
1500
1510
{
1511
http://fabric.com/sets/<set-id>/content/<node-id>?$expand=books($filter=author/any(a: a eq 'Alfred V. Aho')
}

FIG. 15

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/053855

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Amine Ghrab ET AL: "GRAD: On Graph Database Modeling", , 1 February 2016 (2016-02-01), XP055455541, Retrieved from the Internet: URL:https://arxiv.org/abs/1602.00503 [retrieved on 2018-03-01]	1-3,6, 14,15
A	Section 5.2 Example 9 Example 11 Definition 14	4,5,7-13

A	US 9 378 239 B1 (LONG YEJUAN [US] ET AL) 28 June 2016 (2016-06-28) column 3, line 28 - line 33 column 17, line 24 - line 32	1-15

	-/--	

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
9 March 2018	20/03/2018

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Stan, Johann

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/053855

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Spadaccini: "Databases -Constraints and Triggers (N Spadaccini 2009-2010) Databases -Constraints and Triggers 1 / 16", , 31 December 2010 (2010-12-31), XP055422676, Retrieved from the Internet: URL:http://teaching.csse.uwa.edu.au/units/CITS2232/lectures/db-triggers.pdf [retrieved on 2017-11-08] the whole document -----	1-15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/053855

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☒ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-9, 12-15

   Use of a cardinality schema to reject queries that do not satisfy a predefined cardinality constraint.

   ---

2. claims: 10, 11

   Completing a partial query with cardinality constraints.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/053855

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 9378239 B1	28-06-2016	NONE	