(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 2 843 568 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
04.03.2015 Bulletin 2015/10

(51) Int Cl.:
G06F 17/30 $^{(2006.01)}$

(21) Application number: **13461546.7**

(22) Date of filing: **30.08.2013**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME

(71) Applicant: **Pilab S.A.**
50-424 Wroclaw (PL)

(72) Inventor: **Piecko, Krystian**
53-648 Wroclaw (PL)

(74) Representative: **Pawlowski, Adam**
Eupatent.PL
ul. Zeligowskiego 3/5
90-752 Lodz (PL)

Remarks:
Amended claims in accordance with Rule 137(2) EPC.

(54) Computer implemented method for creating database structures without knowledge on functioning of relational database system

(57) A computer implemented method for storing ad hoc relations between previously unrelated database objects assigned to different database structures, the method comprising the steps of: defining at least three database structures: A, B and C wherein there exists a relation between objects of structure A and objects of structure B and wherein there exists a relation between objects of structure B and objects of structure C; filtering data of structure A; accessing structure B using a selected relation between structure A and structure B; storing information about filtering of structure A and information on selected path between structure A and structure B; filtering results obtained from structure B; accessing structure C using a selected relation between structure B and structure C; storing information about filtering of structure B and information on selected path between structure B and structure C.

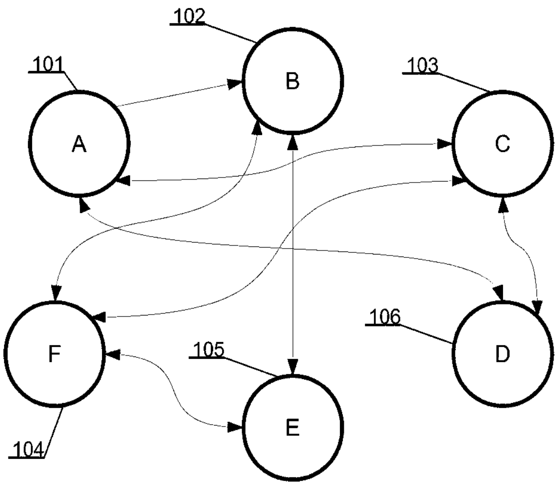


Fig. 1

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The object of the present invention is a computer implemented method for storing ad hoc relations between previously unrelated database objects assigned to different database structures. The method finds its application in data retrieval and processing systems.

[0002] Prior art defines relational databases (source: Wikipedia) as a a database that has a collection of tables of data items, all of which is formally described and organized according to the relational model. The term is in contrast to only one table as the database, and in contrast to other models which also have many tables in one database.

[0003] In the relational model, each table schema should identify a primary column, called the primary key, used for identifying a row. Tables can relate by using a foreign key that usually points to the primary key of another table. The relational model offers various levels of refinement of the table relations called database normalization. (See Normalization below.) The database management system (DBMS) of a relational database is called an RDBMS, and is the software of a relational database.

[0004] When creating a database, usually the designed uses separate tables for different types of entities, for example: users, employees, customers, orders, items etc. What is needed further are relationships between these tables. For instance, customers make orders, and orders contain items. These relationships need to be represented in the database.

[0005] There are several types of database relationships, such as: One to One Relationships, One to Many and Many to One Relationships, Many to Many Relationships, Self Referencing Relationships.

[0006] These relations have to be hardcoded by the database designer, which is quite limiting in large databases, where very high number of relations between data structures may exist. Each change in relations defined in the database required deep interference with database schema and database processes. Therefore, it cannot be executed by a regular end user of a database system.

[0007] Modifications in complex database schemas are very demanding as they need a lot of tests, field trials and experienced database engineers to design and implement.

[0008] SQL provides for creation of Views. A View may be compared to a virtual table, through which a section of data from one or more tables can be viewed. Views however, do not contain data of their own. A View is stored as a SELECT statement in the database. A View contains rows and columns, just like a real table. The fields in a View are fields from one or more real tables in the database.

[0009] Therefore, in order to create ad-hoc relation between objects of different tables, one may create a View and store such view in order to make it available for any other user of the database system.

[0010] Views create the appearance of a table, but the DBMS must still translate queries against the view into queries against the underlying source tables. If the view is defined by a complex, multi-table database query then simple queries on the views may take considerable time. This results in significant performance decrease.

[0011] Additionally, views are new database objects, which means they occupy space and operational memory.

[0012] RDBMS often provides a capability of stored procedures. Stored procedures allow coupling set-based power of SQL with iterative and conditional processing control of code development.

[0013] Drawbacks of stored procedures include limited portability (especially when moving from one database type to another; sometimes even between different versions of the same database), any data errors in handling stored procedures are not generated until runtime. Additionally, too much overhead is incurred from maintaining Stored Procedures that are not complex enough. As a result, simple SELECT statements should not be executed as stored procedures.

[0014] With SQL, there is a possibility to copy information from one table into another. The SELECT INTO statement copies data from one table and inserts it into a new table. This creates a new physical table (PT) that represents data of other tables at the moment of creation of the new table (PT). If after executing SELECT INTO there are created new relations, the new table (PT) will not reflect it.

[0015] Taking into account the foregoing prior art, there is a need to design and implement an efficient storing of ad hoc relations between previously unrelated database objects assigned to different database structures.

[0016] The first object of the present invention is a computer implemented method for storing ad hoc relations between previously unrelated database objects assigned to different database structures, the method comprising the steps of: defining at least three database structures: A, B and C wherein there exists a relation between objects of structure A and objects of structure B and wherein there exists a relation between objects of structure B and objects of structure C; filtering data of structure A; accessing structure B using a selected relation between structure A and structure B; storing information about filtering of structure A and information on selected path between structure A and structure B; filtering results obtained from structure B; accessing structure C using a selected relation between structure B and structure C; storing information about filtering of structure B and information on selected path between structure B and structure C.

[0017] Preferably, the information stored is kept in a form of an XML file.

[0018] Preferably, the XML file includes information related to selected structures, selected relations and selected filters.

[0019] Preferably, the steps of filtering, accessing and storing are executed with respect to a further data structure D related to the data structure C.

[0020] Preferably, the database is a database comprising: a first data structure, stored in the memory, comprising a definition of at least one data set wherein each data set comprises a data set identifier and logically holds data objects of the same type; a second data structure, stored in the memory, comprising definitions of properties of objects wherein each property comprises an identifier of the property and an identifier of a set, from the first data structure, the property is assigned to; a third data structure, stored in the memory, comprising definitions of objects wherein each object comprises an identifier and an identifier of a set, from the first data structure, the object is assigned to; a fourth data structure, stored in the memory, comprising definitions of properties of each object wherein each property of an object associates a value with an object, from the third data structure, and a property of the set, from the second data structure, the object is assigned to; a fifth data structure, stored in the memory, comprising definitions of relations wherein each relation comprises an identifier of the relation; and a sixth data structure, stored in the memory, for storing definitions of relations between objects wherein each objects relation associates a relation, from the fifth data structure, to two objects from the third data structure.

[0021] Another object of the present invention is also computer software comprising program code means for performing all the steps of the computer-implemented method according to the present invention when said program is run on a computer.

[0022] A further object of the present invention is also a computer readable recording medium storing computer-executable instructions performing all the steps of the computer-implemented method according to the present invention when executed on a computer.

[0023] Yet another object of the present invention is a computer implemented system for storing ad hoc relations between previously unrelated database objects assigned to different database structures, the system comprising a database management system configured to: define at least three database structures: A, B and C wherein there exists a relation between objects of structure A and objects of structure B and wherein there exists a relation between objects of structure B and objects of structure C; filter data of structure A; access structure B using a selected relation between structure A and structure B; store information about filtering of structure A and information on selected path between structure A and structure B; filter results obtained from structure B; access structure C using a selected relation between structure B and structure C; store information about filtering of structure B and information on selected path between structure B and structure C.

[0024] These and other objects of the invention presented herein are accomplished by providing a computer-implemented method for storing ad hoc relations between previously unrelated database objects assigned to different database structures. Further details and features of the present invention, its nature and various advantages will become more apparent from the following detailed description of the preferred embodiments shown in a drawing, in which:

FIG. 1 shows an exemplary database structures diagram;
FIG. 2 shows a diagram of processing information by a human brain;
FIG. 3 shows a process of obtaining knowledge;
Fig. 4 presents a method for storing ad-hoc relations;
Fig. 5 shows an exemplary implementation of the method shown in Fig. 4;
Figs. 6 and 7 show a mind-map type database system; and
Fig. 8 presents an exemplary database system, for which the present method has been designed and by which it may be implemented.

[0025] Figure 1 shows an exemplary database structures diagram - a mind map, which can be understood in two different ways:

1. Items A-F are objects;
2. Items A-F are the objects collections (i.e. a data structure).

[0026] The following example shows an important approach of data processing capabilities in a database, especially in a mind map database technology as shown in Figs. 6 and 7. Two distant elements can be linked by a transition to the next closest object/structure. Fig. 1 illustrates that the element 'A' is associated with the element 'E', because the element 'A' is in a direct relation with the element 'D', which has a direct relation with element 'C', which has a direct relation with object 'F', which is directly related with object 'E': A -> D -> C -> F -> E.

[0027] In an automatic analysis, occurrence of the above relationship can be statistically insignificant due to the distance between two objects. However, from the point of view of database data drilling and knowledge extraction, this information may allow to draw appropriate conclusions or to discover a significant piece of information.

[0028] For the purpose of this patent application, the following assumption was made - information is sets of data stored in an RDBMS and it is the element of data processing.

[0029] Knowledge is a collection of such data that the human mind understands, allowing to draw appropriate conclusions and make decisions. In order to make the correct decisions a significant amount of data is required, therefore a decision would be based on knowledge rather than random or partially related pieces of information.

[0030] In the current IT era, the amount of public available data is increasing tremendously over the years. The increasing costs of maintaining such significant volumes of data led to the need for a new data management technology, low-calorie data sets, hereinafter referred to as Big Data. The processing of such information requires the use of statistical analysis and distributed processing. For this purpose mainly Apache Hadoop is used as a data storage technology. The most famous example of Big Data usage is an Internet search engine provided by Google Co. Google's search engine statistical information processing methodology does not allow finding an answer based on the connections to results previously obtained, and thus the knowledge extraction from provided information, unless a human mind-mapping capabilities are added.

[0031] One of the main examples of this kind of issue would be the attempt to find information in a compound system that has a multi-level relationship with a number of related data. For example, identifying the car steering problems that derive from inadequate structure of the car tires.

[0032] The result of a search process focused around a phrase 'steering vibration' gives only a one-dimensional set of answers, in which solution to the problem, most of the time, cannot be discovered without connecting the information. Data analysis is then carried out by the mind of the user and does not result directly from the information model or the search algorithm, thus being limited.

[0033] Processing of information by human brain can be basically summarized in the following steps (as shown in Fig. 2). Step 201 is to obtain basic information using the senses (color, odor, sentence, etc.). Next, step 202 is to categorize information, and adding it to the 'shelf' with a similar elements (all green, all predators etc.). Further, step 203 is to combine information with other 'shelves', and/or other objects.

[0034] In the same manner mind maps are constructed. The process of obtaining knowledge can be summarized as shown in Fig. 3. The first step 301 is to enter an appropriate 'shelf' i.e. information domain. Next, at step 302 there is executed searching for the needed information in the 'shelf'. Further, at step 303 there is executed jumping to another 'shelf' using filtered information and certain connection types. Finally, at step 304 one may return to the second step until a conclusion is made or knowledge is obtained.

[0035] An example of such a query could be a process of finding a tiger:

1. Entrance into big teeth 'shelf';
2. Jumping through connection to wild animals;
3. Filtering the big cats results;
4. Jumping into the 'shelf' with regions
5. Filtering only the Indian results from the 'shelf';
6. Obtaining a result - for instance tiger.

[0036] The example is admittedly exaggerated, and shows that for any information query the human mind can come out with result. The most important suggestion is that using this approach requires the minimum amount of query parameters needed to find appropriate answer from a large dataset.

[0037] By using such approach a knowledgeable person may often find results comprising objects that are somehow related only in the mind of this particular person and which often do not appear related to other persons. Hence, in principle, a large database may logically comprise unlimited number of relations, just as in real world.

[0038] However, relational databases are not programmed to easily extend hardcoded relations. Programming of relations is not a trivial task and potentially is a source of errors, bottlenecks or deadlocks. Frequently, a process such as addition or edit of a record must trigger numerous other operations on related tables and/or records. Therefore, the end user may not easily create new relations in an end product as this requires in depth knowledge of the particular database system design and database programming knowledge.

[0039] The aim of the present invention is to allow storing ad hoc relations between previously unrelated database objects assigned to different database structures, which is very beneficial for knowledge extraction.

[0040] Fig. 4 presents a method for storing ad-hoc relations. The process is designed for databases where there exist at least three database structures A, B and C and wherein there exists a relation between objects of A and objects of B and wherein there exists a relation between objects of B and objects of C. There may of course exist many relations between A and B a user may choose from. Similar assumption applies to B-C relation. The above structure is assumed to have been created in step 401.

[0041] Next, at step 402, a filtering of data is executed on the A structure, for example employees are filtered according to their name. Further, at step 403, structure B is accessed using a selected path (relation), between structure A and structure B. Frequently there will be more that one path to choose from. After that, at step 404, information about filtering (step 402) of the first data structure (A) and information on selected path (step 403) between A and B is stored.

[0042] At this stage, results obtained from data structure B may be filtered 405 and at step 406 data structure C may be accessed using a selected path/relation, between structure B and structure C. Similarly to step 404, at step 407 information about filtering (step 405) of the second data structure (B) and information on selected path (step 406) between B and C is stored.

[0043] Optionally, results obtained from data structure C may be further filtered at step 408. Such filter may be stored at step 409 in case step 408 has been executed.

[0044] Another option that may be executed in a similar manner steps of filtering and moving between data structures, is to apply steps 406, 407 to a further data structure, for example D, related to structure C.

[0045] Fig. 5 shows an exemplary result of the method shown in Fig. 4. The result is a definition of a path defining the identified multi-table logical relation called herein a path. This result is embodied in an XML (Extensible Markup Language) file. The file defines several level of hierarchy in order to define a path comprising identification of a set (setToFilter), conditions of filtering of a particular set (conditions). Further the file defines a relation (stepRelation) selected to connect two chosen sets. As can readily bee see there is not any limit with respect to the number of tables (sets) to be used in a cross-querying path. Hence such definitions are very powerful for database designers and users who wish to create complex, ad-hoc logical relations between different data sets.

[0046] The system presented herein is especially useful in databases based on mind maps such as a database disclosed in the co-pending European Patent Application number EP13461516.0 by the same Applicant, which is hereby incorporated in its entirety by reference. In that particular database type it is especially easy to execute a process of finding objects of interest that are related to different data structures because, a relation between objects is present in the OBJECT RELATIONS data structure.

[0047] The following section of the specification presents key part of the Applicant's co-pending European Patent Application number EP13461516.0.

[0048] FIG. 6, corresponding to Fig. 2 of the co-pending application, shows a new database system according to the present invention. In order to perfectly cooperate with mind maps the database system according to the invention has been designed differently than known database systems. The database system comprises six core sets of data and optional sets. The core sets comprise SETS, OBJECTS, COLUMNS, CHARACTERISTICS, RELATIONS and OBJECTS RELATIONS. It has to be noted that the names above are exemplary only and the respective core sets are defined rather by their function within the system than their name.

[0049] The first set of data is called SETS 604, because it is used to logically hold data related to sets of data. The sets of data may be represented on a mind map as nodes. Each entry in the SETS data structure 604 comprises at least a unique identifier 605a and preferably also its name 605. Referring back to example from Fig. 1 there are three SETS, namely COLORS having ID of 1, MATERIALS having ID of 2 and TOOLS having ID of 3. The SETS data structure is a top level structure and does not refer to other data structures but other data structures refer to it as identified by respective arrows between the sets of Fig. 6.

[0050] Each set of data is, as in the real world, characterized by some properties typically called columns. Hence, the second set of data is called COLUMNS 606. A property, called typically a column, is uniquely identified with an identifier ID 607 and is associated with a set, defined in the SETS data structure 604, by means of an identifier herein called SET ID 608. A column also preferably is associated with a name 609. As indicated by an arrow 604a, the COLUMNS data structure logically, directly references the SETS data structure, because it uses the identifiers of sets. If for example each color of the set called COLORS had another property, say RGB value, there could be added an entry comprising the following values: '1', '4', 'RGB'. At this level of the system the types of respective data such as text, integer, BLOB are not considered as their application in the present system is routine work.

[0051] Having defined data structures of SETS and COLUMNS there may be defined objects that will form elements of respective SETS and will have properties defined by the COLUMNS data structure. Objects are held in the OBJECTS 601 data structure. This data structure holds entries uniquely identified with an identifier ID 603 and associated with a set, defined in the SETS data structure 604, by means of an identifier herein called SET ID 602. As indicated by an arrow 601a, the OBJECTS data structure logically, directly references the SETS data structure, because it uses the identifiers of sets.

[0052] The fourth core data structure is a data structure that holds data entries of each property of each object. This data structure has been called CHARACTERISTICS 301 in Fig. 6. This is one of fundamental differences from all known databases where there are rows of data that comprise entries for all columns of a data table. In the present invention each property of an object is stored as a separate entry, which greatly improves scalability of the system and allows for example adding objects properties in real time.

[0053] The CHARACTERISTICS 701 data structure holds entries uniquely identified with an identifier OBJECT ID 702 and is associated with a property, defined in the COLUMNS data structure 606, by means of an identifier herein called COLUMNID 703. Further each entry in the CHARACTERISTICS data structure, comprises a value of the given property of the particular object. As indicated by respective arrows originating from sources A and B, the CHARACTERISTICS data structure 701 logically, directly references the COLUMNS data structure and the OBJECTS data structure, because it uses the identifiers from the respective data structures.

[0054] The fifth core data structure, of the databases system according to the present invention, is designed to hold data regarding relations present in the database. This data structure has been called herein RELATIONS 705. This is a very simple structure and in principle holds an identifier of a relation ID 707 and preferably also holds the textual description of the relation i.e. a NAME 706. As indicated by an arrow 705a, the RELATIONS data structure logically, directly references downwards the OBJECTS RELATIONS data structure, because the OBJECTS RELATIONS use the identifiers of the relations.

[0055] The last core data structure of the present invention is the mentioned OBJECTS RELATIONS data structure 708. This data structure is designed to provide mapping between a relation from the RELATIONS data structure 705 and two objects from the OBJECTS data structure 701. For example the first entry in the OBJECTS RELATIONS data structure 708 defines that the relation having identifier of 1 exists between object having an identifier of 1 and the object having an identifier of 6.

[0056] Optionally a seventh data structure exists in the database system of the present invention. This data structure holds data regarding relations between the respective data sets and in Fig. 7 is called SETS RELATIONS 712. This data structure is designed to provide mapping between a relation from the RELATIONS data structure 705 and two sets from the SETS data structure 604. For example, the first entry in the SETS RELATIONS data structure 712 defines that the relation having identifier of 1 exists between a set having an identifier of 1 and a set having an identifier of 2. Providing an entry in the SETS RELATION data structure 712 between a set having an identifier of 1 and a set having an identifier of 2 as well as between a set having an identifier of 2 and a set having an identifier of 1, allows for creating a bidirectional relation.

[0057] There is also a possibility of self referencing from a given set. For example such case may be present when there is a set of persons and there exists a student - teacher relation between persons assigned to a particular set.

[0058] As described, for example a relational database system of a hundred tables will in the present system be stored in the six above-described data structures. Naturally, most of the data will be kept in the OBJECTS and CHARACTERISTICS data structures.

[0059] As can be seen in the mind-map-type database, objects are directly related by means of object relations.

[0060] Fig. 8 presents an exemplary database system, for which the present method has been designed and by which it may be implemented. The database system comprises a client 801 and a server 802. The client 801 accesses the data and is typically a remote terminal from the server 802 that hosts a database 806 and a database management system 807 responsible for responding to client's queries. The client is typically a computer comprising a memory 803, a processor 804 and a module 805 for executing the method defined in Fig. 4. It will be evident that a suitable communications channel must be established between the client 801 and the server 802.

[0061] It can be easily recognized, by one skilled in the art, that the aforementioned a computer-implemented method for storing ad hoc relations between previously unrelated database objects assigned to different database structures may be performed and/or controlled by one or more computer programs. Such computer programs are typically executed by utilizing the computing resources in a computing device such as personal computers, personal digital assistants, cellular telephones, receivers and decoders of digital television or the like. Applications are stored in non-volatile memory, for example a flash memory or volatile memory, for example RAM and are executed by a processor. These memories are exemplary recording media for storing computer programs comprising computer-executable instructions performing all the steps of the computer-implemented method according the technical concept presented herein.

[0062] While the invention presented herein has been depicted, described, and has been defined with reference to particular preferred embodiments, such references and examples of implementation in the foregoing specification do not imply any limitation on the invention. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader scope of the technical concept. The presented preferred embodiments are exemplary only, and are not exhaustive of the scope of the technical concept presented herein.

[0063] Accordingly, the scope of protection is not limited to the preferred embodiments described in the specification, but is only limited by the claims that follow.

Claims

1. A computer implemented method for storing ad hoc relations between previously unrelated database objects assigned to different database structures, the method comprising the steps of:

    • defining (401) at least three database structures: A, B and C wherein there exists a relation between objects of structure A and objects of structure B and wherein there exists a relation between objects of structure B and objects of structure C;
    • filtering (402) data of structure A;
    • accessing (403) structure B using a selected relation between structure A and structure B;
    • storing (404) information about filtering (402) of structure A and information on selected path (403) between structure A and structure B;
    • filtering results obtained from structure B (405);
    • accessing (406) structure C using a selected relation between structure B and structure C;
    • storing (407) information about filtering (405) of structure B and information on selected path (406) between structure B and structure C.

2. The method according to claim 1 wherein the information stored is kept in a form of an XML file.

3. The method according to claim 2 wherein the XML file includes information related to selected structures, selected relations and selected filters.

4. The method according to claim 1 wherein the steps of filtering, accessing and storing are executed with respect to a further data structure D related to the data structure C.

5. The method according to claim 1 wherein the database is a database comprising:

• a first data structure (604), stored in the memory, comprising a definition of at least one data set wherein each data set comprises a data set identifier and logically holds data objects of the same type;
• a second data structure (606), stored in the memory, comprising definitions of properties of objects wherein each property comprises an identifier of the property and an identifier of a set, from the first data structure (604), the property is assigned to;
• a third data structure (601), stored in the memory, comprising definitions of objects wherein each object comprises an identifier and an identifier of a set, from the first data structure (604), the object is assigned to;
• a fourth data structure (701), stored in the memory, comprising definitions of properties of each object wherein each property of an object associates a value with an object, from the third data structure (601), and a property of the set, from the second data structure (606), the object is assigned to;
• a fifth data structure (705), stored in the memory, comprising definitions of relations wherein each relation comprises an identifier of the relation; and
• a sixth data structure (708), stored in the memory, for storing definitions of relations between objects wherein each objects relation associates a relation, from the fifth data structure (708), to two objects from the third data structure (601).

6. A computer program comprising program code means for performing all the steps of the computer-implemented method according to any of claims 1 - 5 when said program is run on a computer.

7. A computer readable medium storing computer-executable instructions performing all the steps of the computer-implemented method according to any of claims 1 - 5 when executed on a computer.

8. A computer implemented system for storing ad hoc relations between previously unrelated database objects assigned to different database structures, the system comprising a database management system (807) configured to:

• define (401) at least three database structures: A, B and C wherein there exists a relation between objects of structure A and objects of structure B and wherein there exists a relation between objects of structure B and objects of structure C;
• filter (402) data of structure A;
• access (403) structure B using a selected relation between structure A and structure B;
• store (404) information about filtering (402) of structure A and information on selected path (403) between structure A and structure B;
• filter (405) results obtained from structure B;
• access (406) structure C using a selected relation between structure B and structure C;
• store (407) information about filtering (405) of structure B and information on selected path (406) between structure B and structure C.

Amended claims in accordance with Rule 137(2) EPC.

1. A computer implemented method for storing ad hoc relations between previously unrelated database objects assigned to different database structures, the method comprising the steps of:

• defining (401) at least three database structures, the database structures being collections of data objects,: A, B and C wherein there exists a relation between objects of structure A and objects of structure B and wherein there exists a relation between objects of structure B and objects of structure C;
• filtering (402) data objects of the structure A;
• accessing (403) the structure B using a selected relation between the structure A and the structure B;
• storing (404) information about filtering (402) of the structure A and information on selected relation (403) between the structure A and the structure B wherein the stored information allows for formulation of a database query;
• filtering results obtained from the structure B (405);
• accessing (406) the structure C using a selected relation between the structure B and the structure C;
• storing (407) information about filtering (405) of the structure B and information on the selected relation (406) between the structure B and the structure C wherein the stored information allows for formulation of a database query.

2. The method according to claim 1 wherein the information stored is kept in a form of an XML file.

3. The method according to claim 2 wherein the XML file includes information related to selected structures, selected relations and selected filters.

4. The method according to claim 1 wherein the steps of filtering, accessing and storing are executed with respect to a further database structure D related to the database structure C.

5. The method according to claim 1 wherein the database is a database comprising:

• a first data set (604), stored in the memory, comprising a definition of at least one data set wherein each data set comprises a data set identifier and logically holds data objects of the same type;
• a second data set (606), stored in the memory, comprising definitions of properties of objects wherein each property comprises an identifier of the property and an identifier of a set, from the first data set (604), the property is assigned to;
• a third data set (601), stored in the memory, comprising definitions of objects wherein each object comprises an identifier and an identifier of a set, from the first data set (604), the object is assigned to;
• a fourth data set (701), stored in the memory, comprising definitions of properties of each object wherein each property of an object associates a value with an object, from the third data set (601), and a property of the set, from the second data set (606), the object is assigned to;
• a fifth data set (705), stored in the memory, comprising definitions of relations wherein each relation comprises an identifier of the relation; and
• a sixth data set (708), stored in the memory, for storing definitions of relations between objects wherein each objects relation associates a relation, from the fifth data set (708), to two objects from the third data set (601).

6. A computer program comprising program code means for performing all the steps of the computer-implemented method according to any of claims 1 - 5 when said program is run on a computer.

7. A computer readable medium storing computer-executable instructions performing all the steps of the computer-implemented method according to any of claims 1 - 5 when executed on a computer.

8. A computer implemented system for storing ad hoc relations between previously unrelated database objects assigned to different database structures, the system comprising a database management system (807) configured to:

• define (401) at least three database structures, the database structures being collections of data objects,: A, B and C wherein there exists a relation between objects of the structure A and objects of the structure B and wherein there exists a relation between objects of the structure B and objects of the structure C;
• filter (402) data objects of the structure A;
• access (403) the structure B using a selected relation between the structure A and the structure B;
• store (404) information about filtering (402) of the structure A and information on selected relation (403) between the structure A and the structure B, wherein the stored information allows for formulation of a database query;
• filter (405) results obtained from the structure B;
• access (406) the structure C using a selected relation between the structure B and the structure C;
• store (407) information about filtering (405) of the structure B and information on selected relation (406) between the structure B and the structure C wherein the stored information allows for formulation of a database query.

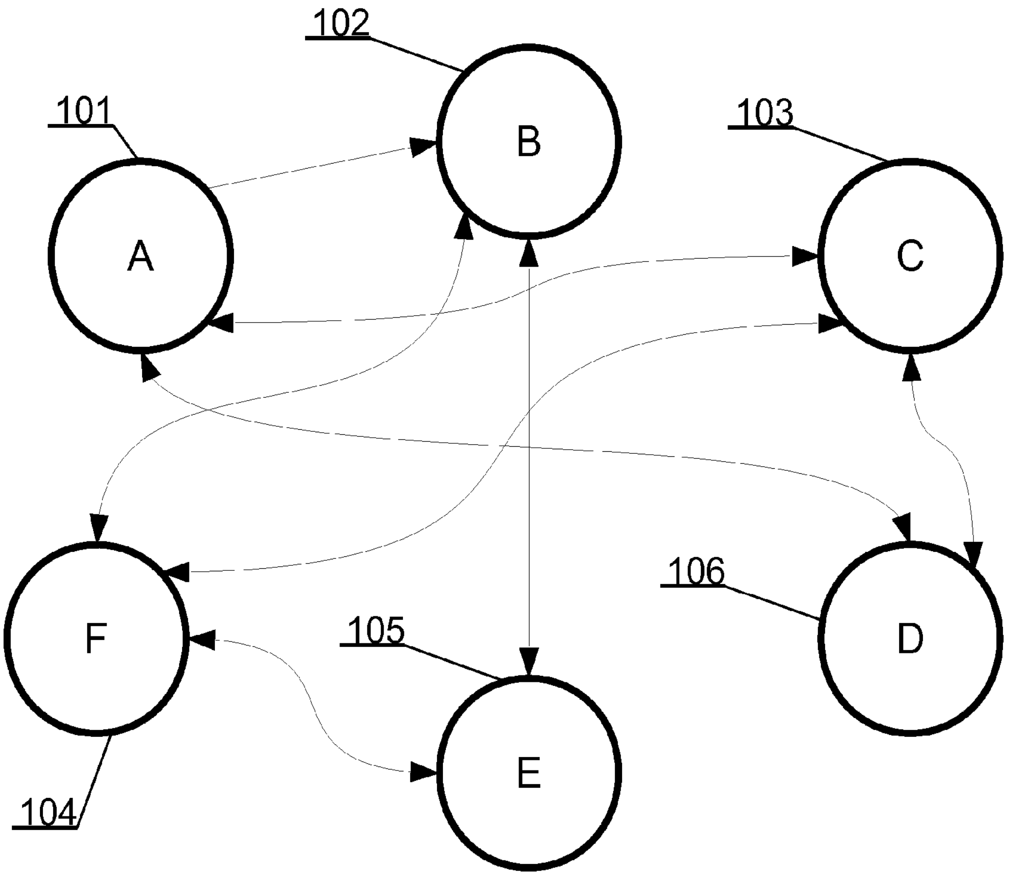


Fig. 1

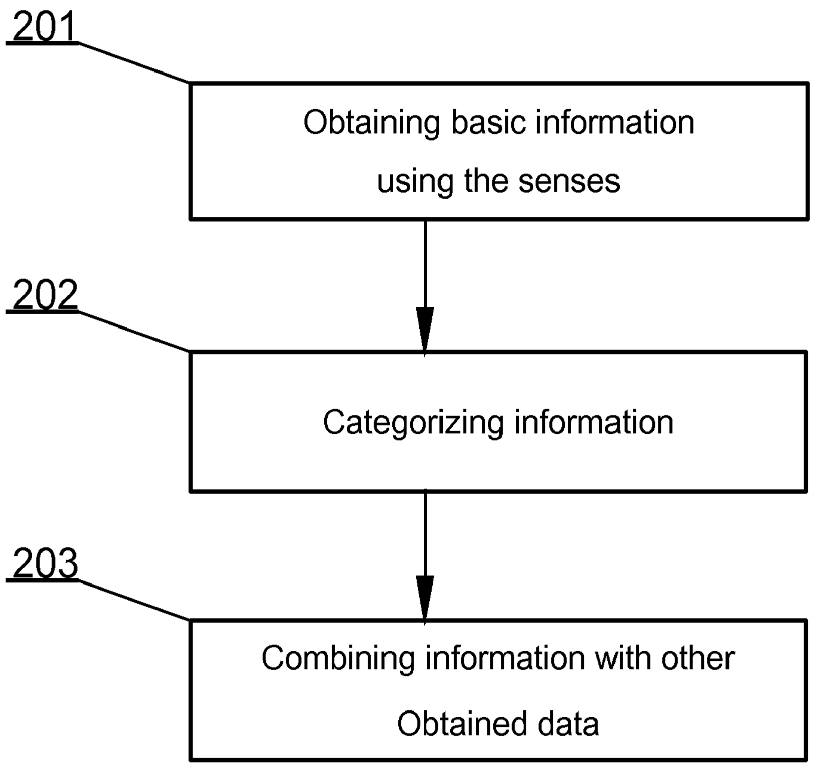

201
Obtaining basic information using the senses
202
Categorizing information
203
Combining information with other Obtained data

Fig. 2

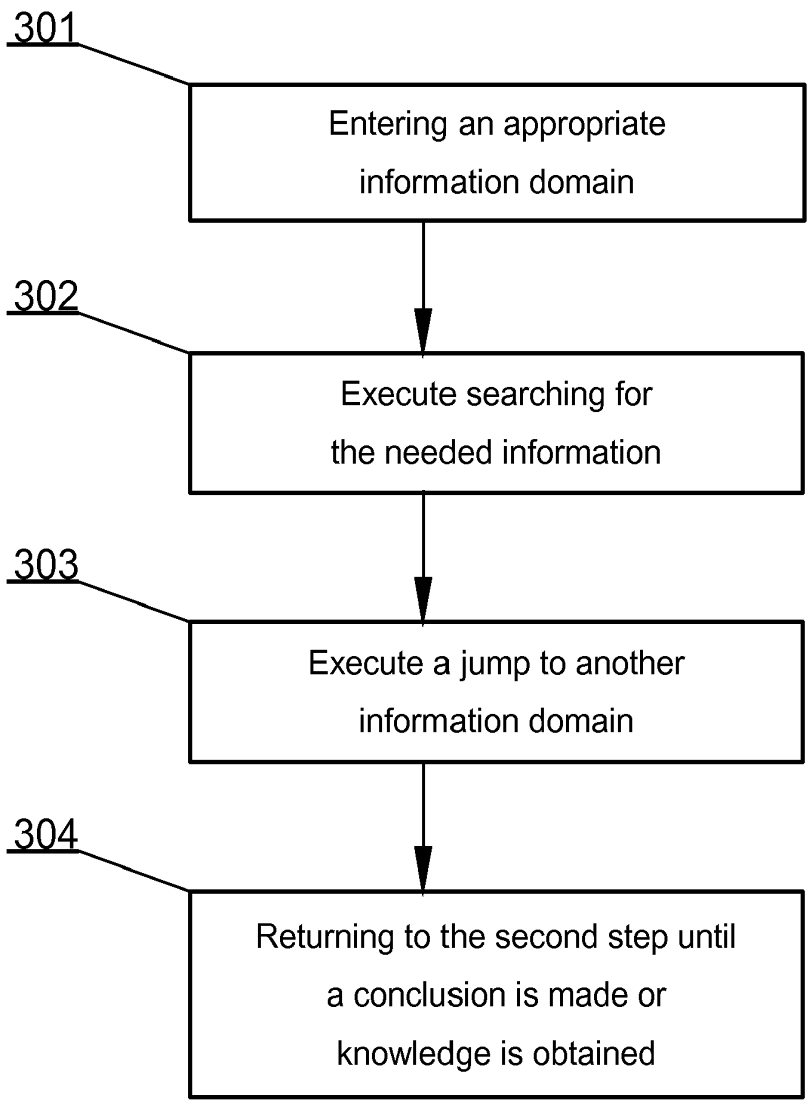

301
Entering an appropriate information domain
302
Execute searching for the needed information
303
Execute a jump to another information domain
304
Returning to the second step until a conclusion is made or knowledge is obtained

Fig. 3

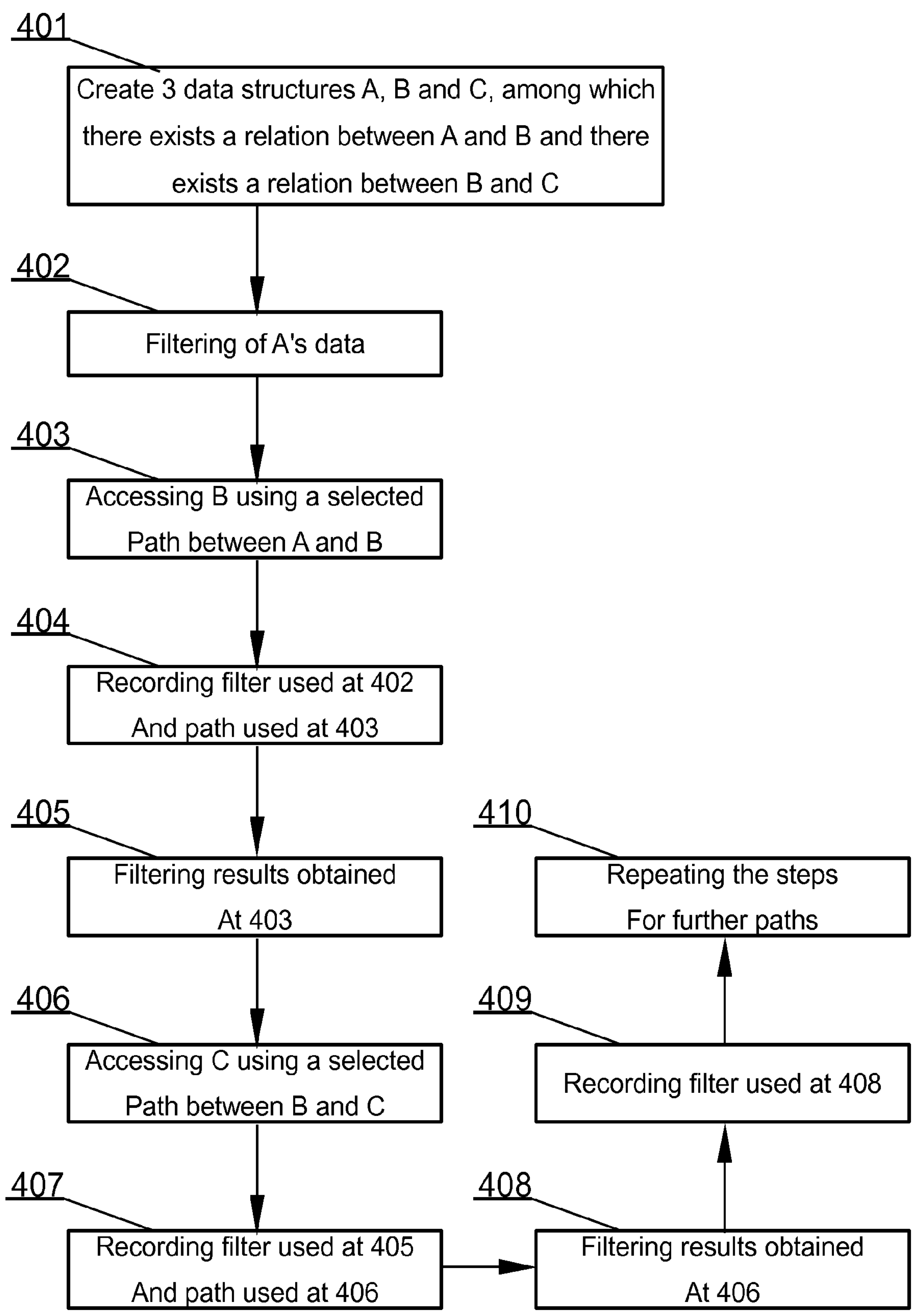
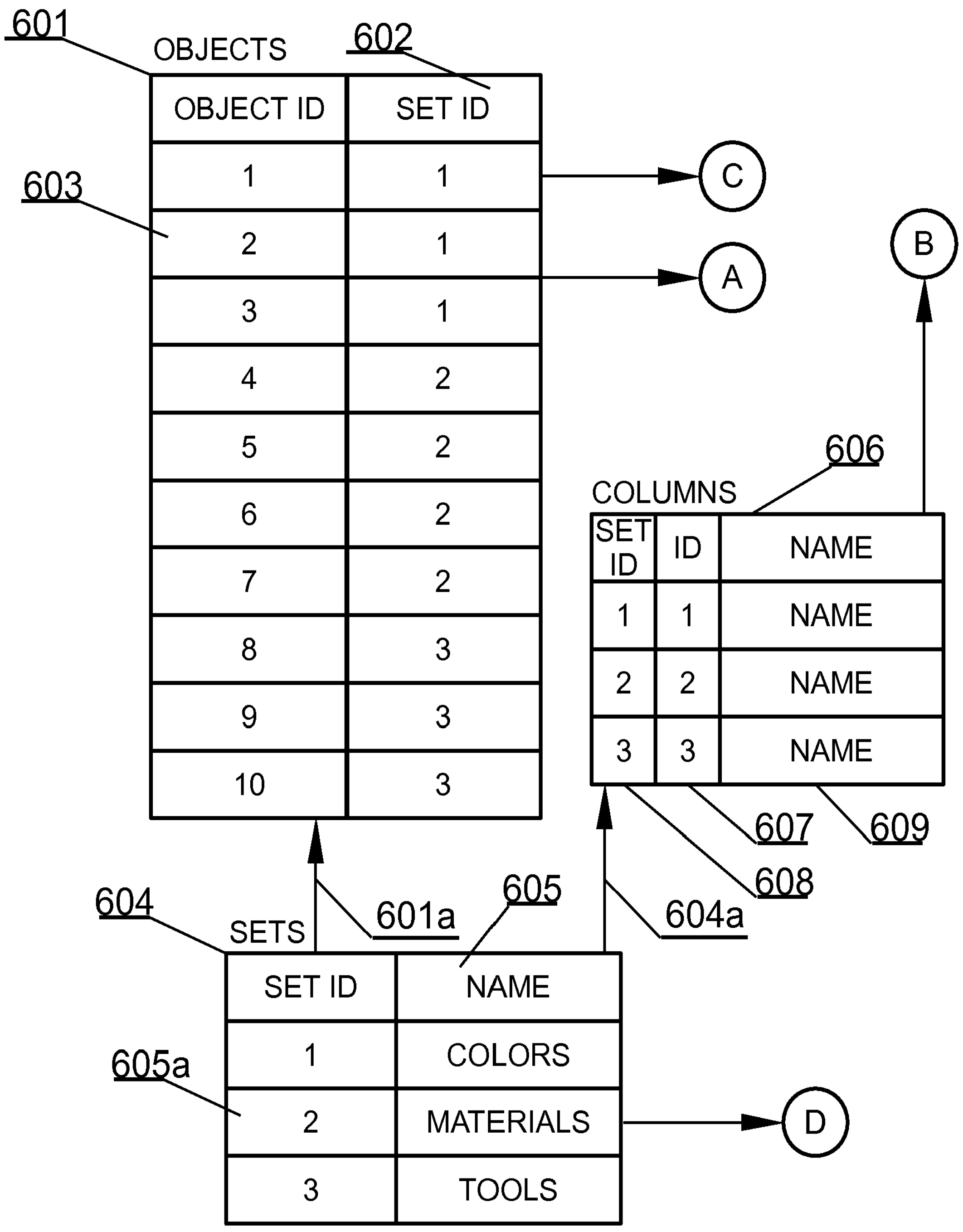

401
Create 3 data structures A, B and C, among which there exists a relation between A and B and there exists a relation between B and C
402
Filtering of A's data
403
Accessing B using a selected Path between A and B
404
Recording filter used at 402 And path used at 403
405
Filtering results obtained At 403
406
Accessing C using a selected Path between B and C
407
Recording filter used at 405 And path used at 406
408
Filtering results obtained At 406
409
Recording filter used at 408
410
Repeating the steps For further paths

Fig. 4

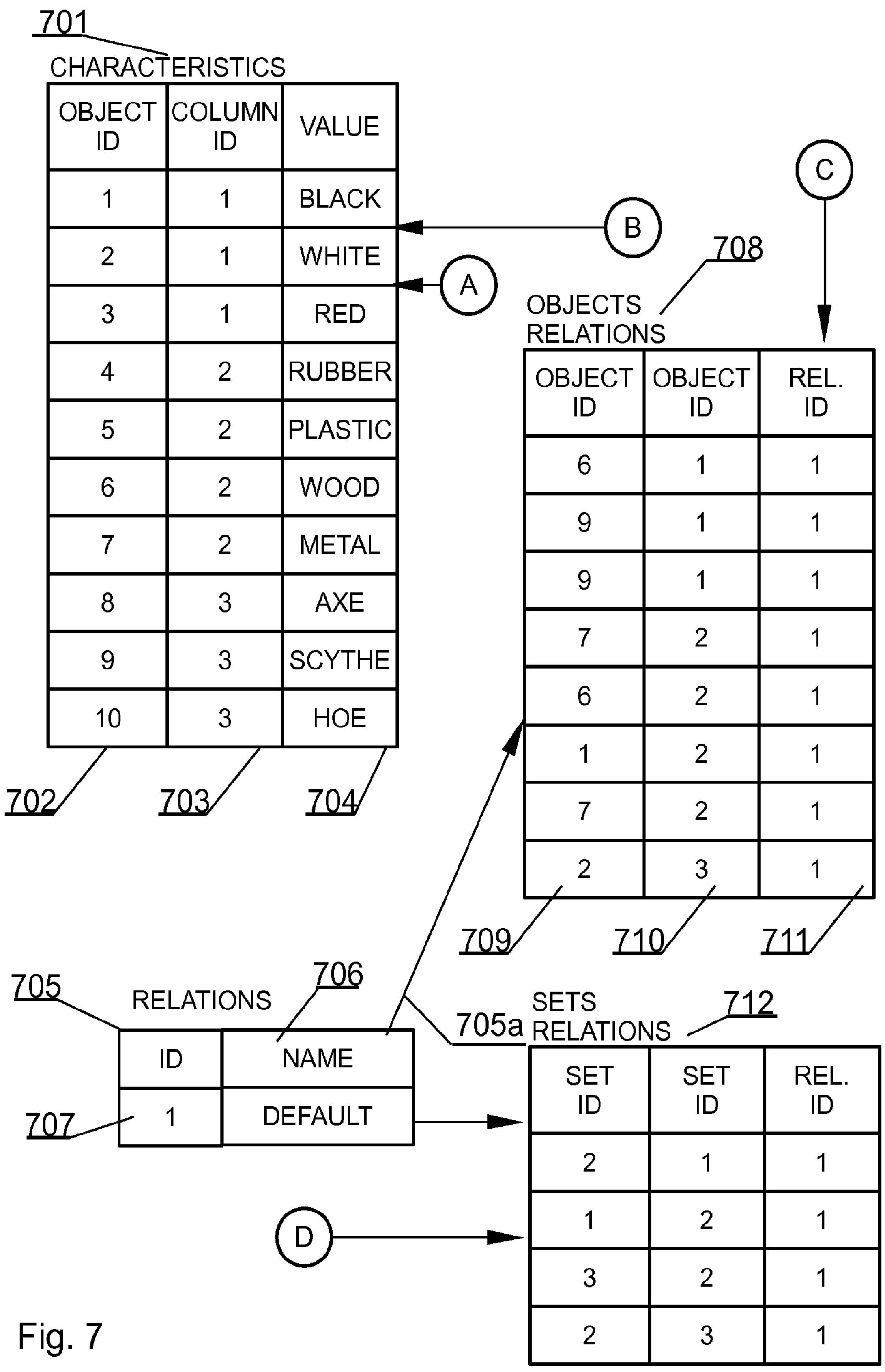
```
<path>
        <step>
                <setToFilter>A</setToFilter>
                <conditions>
                        <condition> (…) </condition>
                        (…)
                        (…)
                        (…)
                </conditions>
        </step>
        <stepRelation = 'Relation_A'>
                <step>
                        <setToFilter>B</setToFilter>
                        <conditions>
                                <condition> (…) </condition>
                                (…)
                                (…)
                                (…)
                        </conditions>
                </step>
        </stepRelation>
        <stepRelation = 'Relation_C'>
                <step>
                        <setToFilter>C</setToFilter>
                        <conditions>
                                <condition> (…) </condition>
                                (…)
                                (…)
                                (…)
                        </conditions>
                </step>
        </stepRelation>
</path>
```

Fig. 5



601
OBJECTS
602
OBJECT ID
SET ID
1
1
2
1
3
1
4
2
5
2
6
2
7
2
8
3
9
3
10
3
603
C
A
B
606
COLUMNS
SET ID
ID
NAME
1
1
NAME
2
2
NAME
3
3
NAME
607
609
608
604a
604
SETS
601a
605
SET ID
NAME
1
COLORS
605a
2
MATERIALS
3
TOOLS
D

Fig. 6



CHARACTERISTICS

OBJECT ID	COLUMN ID	VALUE
1	1	BLACK
2	1	WHITE
3	1	RED
4	2	RUBBER
5	2	PLASTIC
6	2	WOOD
7	2	METAL
8	3	AXE
9	3	SCYTHE
10	3	HOE

OBJECTS RELATIONS

OBJECT ID	OBJECT ID	REL. ID
6	1	1
9	1	1
9	1	1
7	2	1
6	2	1
1	2	1
7	2	1
2	3	1

RELATIONS

ID	NAME
1	DEFAULT

SETS RELATIONS

SET ID	SET ID	REL. ID
2	1	1
1	2	1
3	2	1
2	3	1

Fig. 7

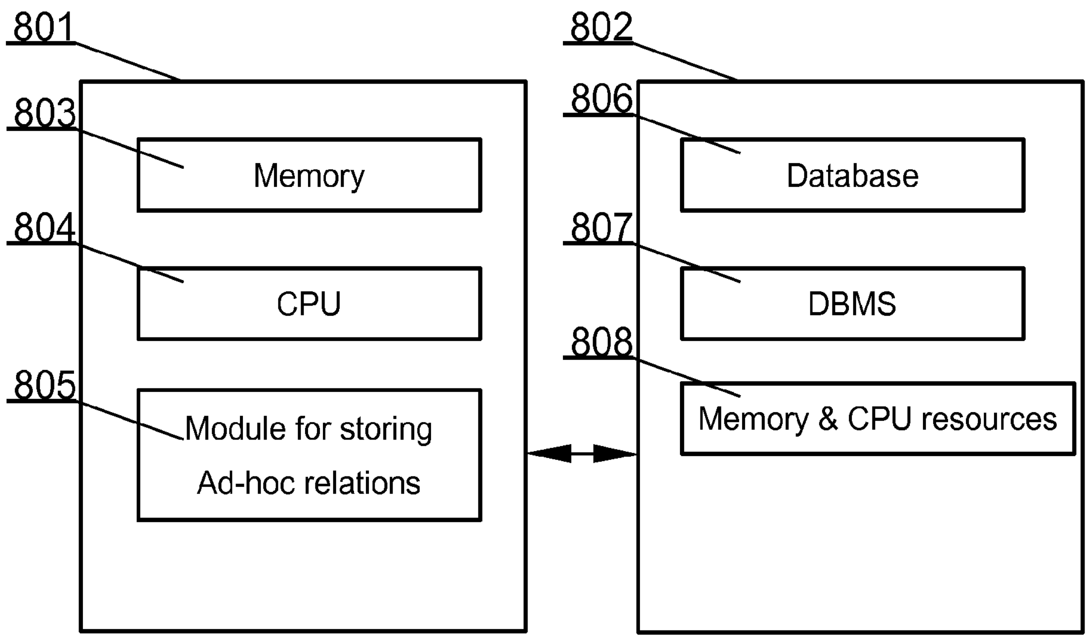

801
802
803
Memory
804
CPU
805
Module for storing
Ad-hoc relations
806
Database
807
DBMS
808
Memory & CPU resources

Fig. 8

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 13 46 1546

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 2007/198557 A1 (CHING AARON K Y [US] ET AL) 23 August 2007 (2007-08-23) * figures 3,4,7A,7B * * paragraph [0018] - paragraph [0019] * * paragraph [0034] - paragraph [0050] * * claim 15 * -----	1-8	INV. G06F17/30
A	US 6 947 945 B1 (CAREY MICHAEL JAMES [US] ET AL) 20 September 2005 (2005-09-20) * figures 1,2,3,4 * * column 3, line 5 - column 3, line 10 * * column 4, line 55 - column 5, line 13 * * column 8, line 5 - column 8, line 7 * -----	1-3	
A	US 6 192 371 B1 (SCHULTZ THOMAS ALAN [US]) 20 February 2001 (2001-02-20) * column 5, line 10 - column 6, line 56 * * figures 4,5,8,9,10 * -----	5	
A	US 6 105 035 A (MONGE DARYL LEE [US] ET AL) 15 August 2000 (2000-08-15) * column 3, line 38 - column 4, line 45 * * figures 5,7,9,10,11,11a * -----	5	TECHNICAL FIELDS SEARCHED (IPC) G06F

The present search report has been drawn up for all claims

2

Place of search	Date of completion of the search	Examiner
Munich	26 March 2014	Yotova, Polina

EPO FORM 1503 03.82 (P04C01)

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
&: member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 13 46 1546

26-03-2014

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007198557 A1	23-08-2007	NONE	
US 6947945 B1	20-09-2005	NONE	
US 6192371 B1	20-02-2001	NONE	
US 6105035 A	15-08-2000	NONE	

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

REFERENCES CITED IN THE DESCRIPTION

Patent documents cited in the description

- EP 13461516 A **[0046] [0047]**