Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2588922 C 2016/02/09

(11)(21) 2 588 922

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2005/12/02
(87) Date publication PCT/PCT Publication Date: 2006/07/13
(45) Date de délivrance/Issue Date: 2016/02/09
(85) Entrée phase nationale/National Entry: 2007/05/28
(86) N° demande PCT/PCT Application No.: US 2005/043563
(87) N° publication PCT/PCT Publication No.: 2006/073633
(30) Priorités/Priorities: 2004/12/30 (US60/640,469); 2005/11/09 (US11/271,702)

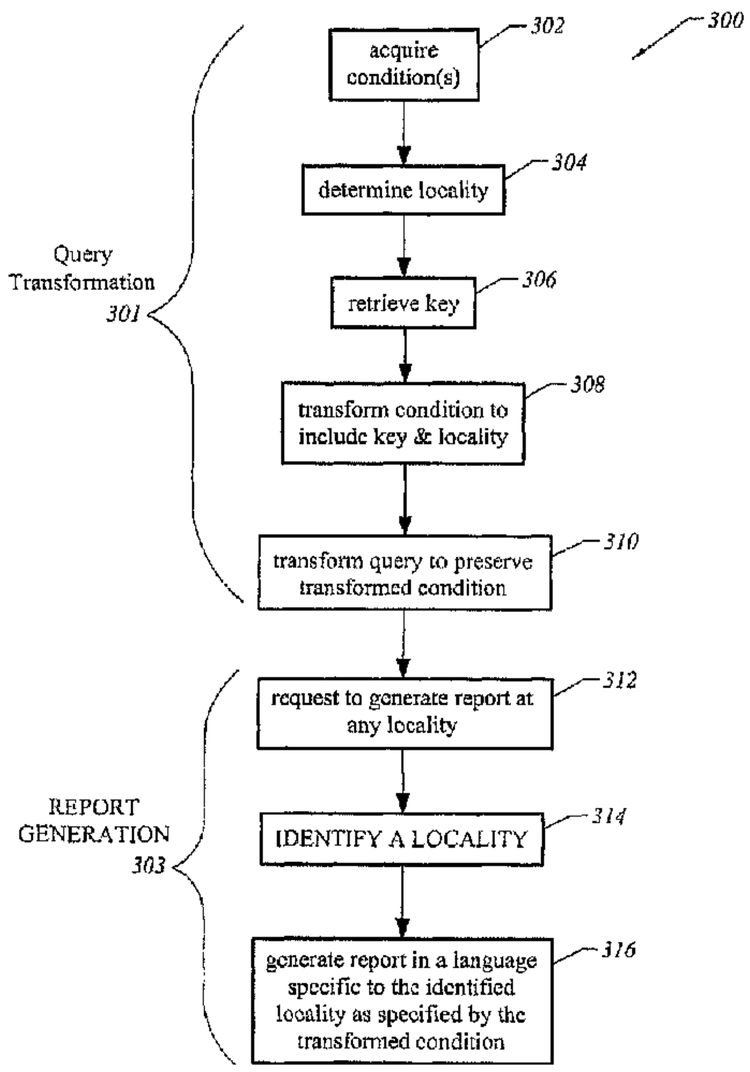
(51) Cl.Int./Int.Cl. *G06F 17/30* (2006.01)
(72) Inventeur/Inventor:
POLO-MALOUVIER, RICARDO, FR

(73) Propriétaire/Owner:
SAP FRANCE, FR

(74) Agent: GOWLING LAFLEUR HENDERSON LLP

(54) Titre : SUPPORT LISIBLE PAR ORDINATEUR, PROCEDE ET APPAREIL DE CONSERVATION DES CONDITIONS DE FILTRAGE POUR INTERROGER DES SOURCES DE DONNEES MULTILINGUES SELON DIVERSES LOCALISATIONS LORS DE LA REGENERATION D'UN COMPTE REND

(54) Title: COMPUTER READABLE MEDIUM, METHOD AND APPARATUS FOR PRESERVING FILTERING CONDITIONS TO QUERY MULTILINGUAL DATA SOURCES AT VARIOUS LOCALES WHEN REGENERATING A REPORT



(57) Abrégé/Abstract:
A computer readable medium, system, apparatus and method are disclosed for generating and regenerating query results in reports whereby conditions for filtering a query against multilingual databases are preserved independent of language and/or locale. According to one embodiment of the present invention, a computer readable medium includes executable instructions to specify a language-dependent value for filtering query results during a query. Other executable instructions are included to associate a first locale to the language-dependent value, determine a key based on the language-dependent value and on the first locale, and generate a transformed query, which can include a locale variable configured to indicate a second locale for regenerating the query results to form regenerated query results for the second locale using another language-dependent value.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 July 2006 (13.07.2006)

PCT

(10) International Publication Number
WO 2006/073633 A2

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

(21) International Application Number:
PCT/US2005/043563

(22) International Filing Date:
2 December 2005 (02.12.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/640,469	30 December 2004 (30.12.2004)	US
11/271,702	9 November 2005 (09.11.2005)	US

(71) Applicant: **BUSINESS OBJECTS, S.A.** [FR/US]; 157-159, Rue Anatole, F-92309 Levallois-perret (FR).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **POLO-MALOUVIER, Ricardo** [FR/FR]; 4, Rue Pasteur, F-92300 Levallois-perret (FR).

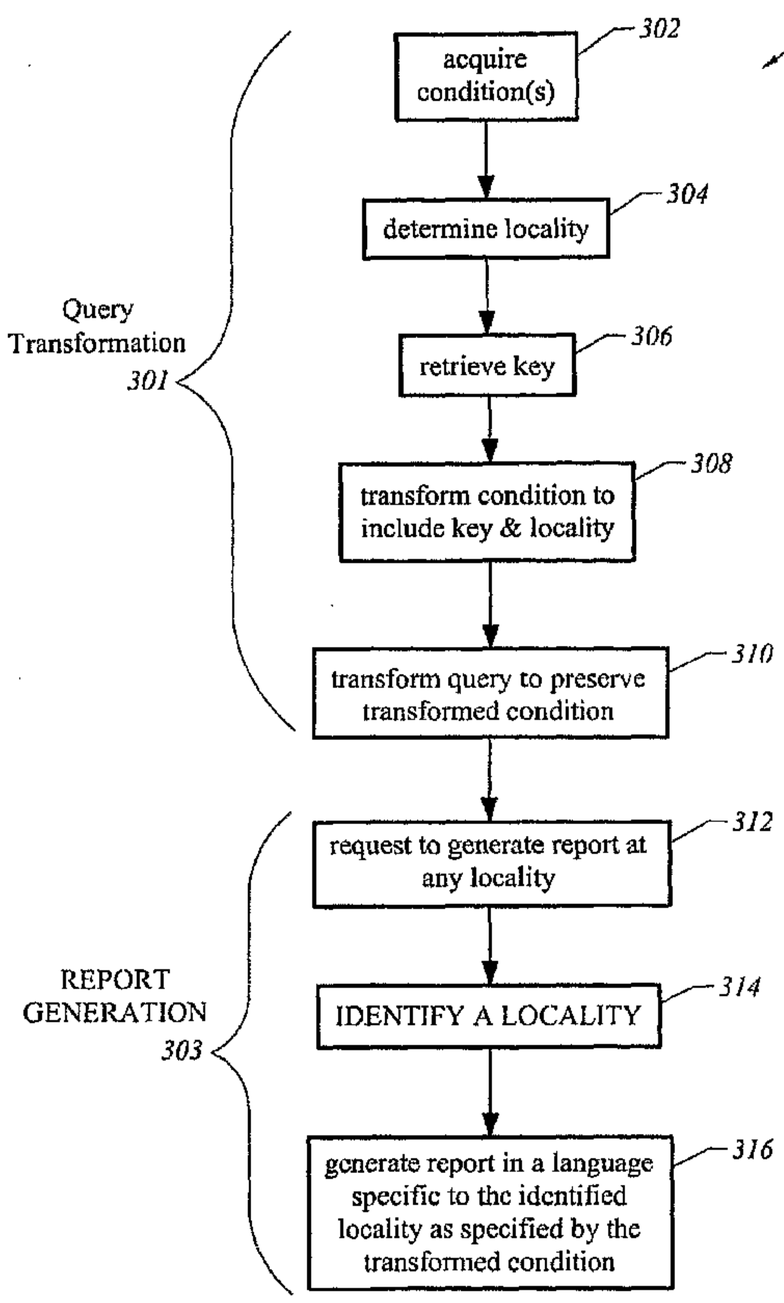
(74) Agent: **BACKUS, Ken, R.**; Cooley Godward, LLP, Five Palo Alto Square, 3000 El. Camino Road, Palo Alto, California 94306-2155, (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: COMPUTER READABLE MEDIUM, METHOD AND APPARATUS FOR PRESERVING FILTERING CONDITIONS TO QUERY MULTILINGUAL DATA SOURCES AT VARIOUS LOCALES WHEN REGENERATING A REPORT



(57) Abstract: A computer readable medium, system, apparatus and method are disclosed for generating and regenerating query results in reports whereby conditions for filtering a query against multilingual databases are preserved independent of language and/or locale. According to one embodiment of the present invention, a computer readable medium includes executable instructions to specify a language-dependent value for filtering query results during a query. Other executable instructions are included to associate a first locale to the language-dependent value, determine a key based on the language-dependent value and on the first locale, and generate a transformed query, which can include a locale variable configured to indicate a second locale for regenerating the query results to form regenerated query results for the second locale using another language-dependent value.

WO 2006/073633 A2

WO 2006/073633 A2

Declaration under Rule 4.17:

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

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

COMPUTER READABLE MEDIUM, METHOD AND APPARATUS FOR PRESERVING FILTERING CONDITIONS TO QUERY MULTILINGUAL DATA SOURCES AT VARIOUS LOCALES WHEN REGENERATING A REPORT

[0001]

BRIEF DESCRIPTION OF THE INVENTION

[0002] This invention relates generally to the generation of reports in a computer network. More particularly, this invention relates to a technique for preserving conditions (e.g., semantics) of a query to filter multilingual databases based on a language and/or locale specific to regeneration of query results, or reports.

BACKGROUND OF THE INVENTION

[0003] Multilingual data sources, such as multilingual databases, generally store translated label values representing words in different languages, such as English, German or French. Multilingual data schemas have been developed to store, relate and access those translated label values depending on the language of interest. A "translated label value," or "TLV," refers generally to the data representing a specific translated word from one of many languages, whereby a common meaning usually relates a collection of translated label values over a number of different languages. Examples of common multilingual data structures are depicted in FIGs. 1A to 1D.

[0004] FIG. 1A shows a portion of common data structure 100 for relating multilingual data for generating reports. Data structure 100 contains "duplicate rows" where each of the duplicated rows relates a specific translated label to a specific concept, such as a product identifier ("ID"). Column 102 includes a common characteristic, such a product ID, column 104 indicates a language to which a translated label belongs, and column 106

contains the translated labels (or words), such as "car," "voiture," and "wagen," all of which relate to a common meaning. As such, these translated labels of duplicate rows relate to a group of common product IDs, such as groups 108 and 110. To find a specific label for a query, specific values in both columns 102 and 104 are required to determine a product name ("Prod_Name"). FIG. 1B shows a portion of another common data structure 120 for relating multilingual data in a manner similar to that of FIG. 1A. Data structure 120, however, contains "duplicate columns" where each of the duplicated columns relates a specific translated label to a product ID as a specific concept. Specifically, column 122 includes product IDs as a common characteristic, and duplicated columns 124 indicate product names having translated labels expressed in different languages. As shown, columns 124a, 124b, and 124c include translated labels in English, French, and German, respectively, and rows 228 and 230 identify specific product IDs, such as P0010 and P0020, respectively.

[0005] FIG. 1C depicts a portion of another traditional multilingual data structure that includes at least two separate data structures 140a and 140b, each of which contains labels of one specific language. In this case, table 140a is described as an English table 146a for words used in, for example, the United States ("US"), whereas table 140b is a French table 146b for words used in, for example, France. Columns 142a and 142b include product IDs and columns 144a and 144b includes respective translated labels, but in different languages. FIG. 1D illustrates a portion of yet another conventional multilingual data structure where a sub-table 170 of translated values is dynamically produced from a base table 160 to improve performance of data access as compared to the data structures shown in FIGs. 1A to 1C. In particular, base table 160 includes product IDs and a specific untranslated value, such as a "price," in columns 162 and 164, respectively. Given product ID P0010, sub-table 170 is generated to include columns 172, 174 and 176 to relate product IDs, language, and product names, respectively.

[0006] While the foregoing data structures and associated multilingual data schemas are functional, there is a common drawback in the implementation of these data structures when reports are generated in different locales, especially where the languages at those locales are also different. Labels, such as product names, are traditionally hard-coded into a query for matching translated labels in a multilingual data source against which a user input queries. For instance, consider that one desires to determine the sales revenue for a given product, such as a product having an English name of "plane." To perform a query on the sales revenue for a product named plane, one traditionally has to design a filtered query so

that only rows matching this product are retrieved from the data source (when implementing a "duplicate rows" schema). When building a filtered query, a user typically uses the product names that are in their own language, rather than either in a foreign language or as represented by some untranslatable attribute, such as a product identifier, that is not readily discernable to a user. Generally, untranslatable attributes are identifiers that are internal to a database and are usually understood only by those skilled in manipulating specialized database programming languages. To build a conventional filtered query for the English-named product plane, a skilled user typically develops a code snippet equivalent to WHERE PRODUCT.PROD_NAME = "plane" to query a multilingual database. The drawback to this approach is that in another locale (or language), the translated label in one language cannot match the user input in another language, and as such, a regenerated report at another locale can lead to improper query results. For example, if a query is coded to only search for French labels of table 100, as defined by a specific language field, LANG "FR," then English words, such as "plane," cannot be used to filter the query when it is rerun, unless the software processes conducting such a query are modified for use with the English language. As such modifications are typically manual in nature, there are additional costs and/or efforts required to implement the data structures of FIGs. 1A to 1D in multilingual database operations, such as in report generation.

[0007] In view of the foregoing, it would be highly desirable to provide an improved technique for generating reports from data maintained in multilingual data sources. In particular, it would be highly desirable to generate reports in a language at a locale different from that where the original query was formed, without manually modifying queries and/or software programs to do so.

SUMMARY OF THE INVENTION

[0008] A computer readable medium, system, apparatus and method are disclosed for generating and regenerating query results in reports whereby conditions for filtering queries against multilingual databases are preserved independent of language and/or locale. According to one embodiment of the present invention, a computer readable medium includes executable instructions to regenerate results of a query in a language other than one used to initially generate the results of the query. The computer readable medium includes executable instructions to specify a language-dependent value of a plurality of language-dependent values for filtering query results in accordance with the one language-dependent

value. Other executable instructions are included to associate a first locale to the language-dependent value, determine an language-independent key based on the language-dependent value and on the first locale, and generate a transformed query with a locale variable that is configured to indicate a second locale for regenerating the query results at or for the second locale. For example, the query results can be in a first language and the regenerated query results can be in a second language. The language-dependent value and another language-dependent value can each be a string.

[0009] According to another embodiment of the present invention, a computer readable medium includes executable instructions to specify query parameters, which are configured to formulate a query. Also, the computer readable medium includes executable instructions to specify a condition identifying a first translated label value associated with a first language. Other executable instructions are included to transform the condition to form a translated condition as a function of a locale variable, and implement the translated condition with the query to form a transformed query that is language-independent prior to generating a report. The transformed query is configured to automatically regenerate the results of the query as part of the report in a second language.

[0010] According to yet another embodiment of the present invention, an apparatus is disclosed for preserving filtering conditions to regenerate query results using multilingual data sources for various locales. The apparatus includes a locale manager configured to provide at least a locale indicator identifying a first language and a multilingual query engine. The multilingual query engine is configured to access data sources each maintaining data in multiple languages. Notably, the multilingual query engine is configured further to accept query parameters of a query, to acquire a condition for filtering the query to provide query results in terms of the condition, and to transform the query into a transformed query. The transformed query is configured to provide the query results in any language as determined by one of a plurality of locale indicators. In a specific embodiment, the apparatus also includes a report engine configured to generate a report including the query results in a language specified by one of the plurality of locale indicators.

[0010A] In accordance with one aspect of the invention there is provided a method of regenerating results of a query in a language other than one used to initially generate the results of the query. The method involves specifying a language-dependent value of a

plurality of language-dependent values for filtering query results in accordance with the language-dependent value, and associating a first locale to the language-dependent value. The method further involves determining a key based on the language-dependent value and on the first locale, the key being language-independent. The method further involves generating a transformed query to include the determined key and a locale variable configured to indicate a second locale for regenerating the query results to form regenerated query results for the second locale using another language-dependent value of the plurality of language-dependent values.

[0010B] The query results may be in a first language and the regenerated query results may be in a second language, and the language-dependent value and the another language-dependent value may each be a string.

[0010C] The method may involve setting the locale variable to represent the second locale, and forming the regenerated query results for presentation for the second locale.

[0010D] An object may facilitate regenerating the query results by automatically forming the transformed query.

[0010E] Determining the key may involve searching a multilingual data source using the language-dependent value and a first locale indicator representing the first locale, and retrieving the key.

[0010F] Forming the regenerated query may involve searching the multilingual data source using the key and a second locale indicator representing the second locale to retrieve the another language-dependent value.

[0010G] Searching the multilingual data source using said key and said second locale indicator may be done without matching the language-dependent value against the data source.

[0010H] The method may involve detecting the locale variable is set as the second locale, and supplying a prompt to a user in a language associated with the second locale to provide a user input including the another language-dependent value. The prompt may facilitate receiving the another language-dependent value dynamically.

[0010I] Supplying the prompt may involve retrieving a list of selections in the language from which the another language-dependent value is selected.

[0010J] Each of the first and the second locales may be unique to a user and the method may further involve specifying a language in which to regenerate query results regardless of geographical location.

[0010K] In accordance with another aspect of the invention there is provided a method of implementing multilingual objects to regenerate the results of queries in a language other than one used to initially generate the queries. The method involves specifying query parameters configured to formulate a query, specifying a condition identifying a first translated label value associated with a first language, and transforming the condition to form a translated condition as a function of a locale variable. The method further involves implementing the translated condition with the query to form a transformed query that is language-independent prior to generating a report. The transformed query is configured to automatically regenerate the results of the query as part of the report in a second language.

[0010L] The method may involve constructing a multilingual object configured to form associations among a base object specifying a number of language-independent keys, a locale field to specify a field in a multilingual data source, and two or more translated label values each of which is associated with a different language.

[0010M] Transforming the condition may further involve using the multilingual object to retrieve a key from the number of language-independent keys that are stored in the multilingual data source, the locale field and the first translated label value being used to query the multilingual data source.

[0010N] Implementing the translated condition may further involve defining the transformed query in terms of the key and a locale variable.

[0010O] In accordance with another aspect of the invention, there is provided an apparatus for preserving filtering conditions to regenerate query results using multilingual data sources for various locales. The apparatus includes a locale manager configured to provide at least a locale indicator identifying a first language, and a multilingual query engine configured to access data sources each maintaining data in multiple languages, and configured further to accept query parameters of a query, acquire a condition for filtering the query to provide query results in terms of the condition, and transform the query into a transformed query. The transformed query is configured to provide the query results in any language as determined by one of a plurality of locale indicators.

[0010P] The apparatus may include a report engine configured to generate a report including the query results in a language specified by the one of the plurality of locale indicators.

[0010Q] The apparatus may further include a data store for maintaining multilingual objects, at least one of which defines associations among the condition, any of the plurality of locale indicators, and at least a language-independent key.

[0010R] The multilingual query engine may be further configured to retrieve the key from the data sources using the locale indicator and the condition.

[0010S] The multilingual query engine may be further configured to form the transformed query by associating the query with the key and a variable capable of having a value equivalent to any of the plurality of locale indicators.

[0010T] In accordance with another aspect of the invention, there is provided a method of regenerating results of a query in a language other than one used to initially generate the results of the query. The method involves specifying a language-dependent value of a plurality of language-dependent values for filtering query results in accordance with the language-dependent value, associating a first locale to the language-dependent value, and determining a key based on the language-dependent value and on the first locale, the key being common across tables storing information in different languages. The method further involves generating a transformed query to include a locale variable configured to indicate a second locale for regenerating the query results to form regenerated query results for the second locale using another language-dependent value of the plurality of language-dependent values, detecting the locale variable is set as the second locale, and supplying a prompt to a user in a language associated with the second locale to provide a user input including the another language-dependent value, wherein the prompt facilitates receiving the another language-dependent value dynamically. The transformed query is applied to the tables to form the regenerated results. The query results are in a first language and the regenerated query results are in a second language, and the language-dependent value and the another language-dependent value are each a string.

[0010U] The method may further involve setting the locale variable to represent the second locale, and forming the regenerated query results for presentation for the second locale.

[0010V] An object may facilitate regenerating the query results by automatically forming the transformed query.

[0010W] Determining the key may involve searching a multilingual data source using the language-dependent value and a first locale indicator representing the first locale, and retrieving the key.

[0010X] Forming the regenerated query may involve searching the multilingual data source using the key and a second locale indicator representing the second locale to retrieve the another language-dependent value.

[0010Y] Searching the multilingual data source using said key and said second locale indicator may be done without matching the language-dependent value against the data source.

[0010Z] Supplying the prompt may involve retrieving a list of selections in the language from which the another language-dependent value is selected.

[0010AA] Each of the first and the second locales may be unique to a user and the method may further involve specifying a language in which to regenerate query results regardless of geographical location.

[0010AB] In accordance with another aspect of the invention, there is provided a computer system for preserving filtering conditions to regenerate query results using multilingual data sources for various locales. The computer system includes a processor and a memory, and a locale manager configured to provide at least a locale indicator identifying a first language. The system further includes a multilingual query engine configured to access multilingual data sources, and configured further to accept query parameters of a query, acquire a condition for filtering the query to provide query results in terms of the condition, and transform the query into a transformed query for application to the multilingual data sources. The system further includes a data store for maintaining multilingual objects, at least one of which defines associations among the condition, any of the plurality of locale indicators, and at least a language-independent key. The transformed query is configured to provide the query results in any language as determined by one of a plurality of locale indicators. The multilingual query engine is further configured to retrieve the key from the data sources using the locale indicator and the condition. The multilingual query engine is

further configured to form the transformed query by associating the query with the key and a variable capable of having a value equivalent to any of the plurality of locale indicators.

[0010AC] The computer system may further include a report engine configured to generate a report including the query results in a language specified by the one of the plurality of locale indicators.

[0010AD] In accordance with another aspect of the invention, there is provided a computer-readable medium having stored thereon computer code for directing a computer to execute any one of the above methods.

[0010AE] In accordance with another aspect of the invention, there is provided an apparatus for regenerating results of a query in a language other than one used to initially generate the results of the query. The apparatus includes provisions for specifying a language-dependent value of a plurality of language-dependent values for filtering query results in accordance with the language-dependent value, and provisions for associating a first locale to the language-dependent value. The apparatus further includes provisions for determining a key based on the language-dependent value and on the first locale, the key being language-independent, and provisions for generating a transformed query to include the determined key and a locale variable configured to indicate a second locale for regenerating the query results to form regenerated query results for the second locale using another language-dependent value of the plurality of language-dependent values.

[0010AF] The query results may be in a first language and the regenerated query results may be in a second language, and the language-dependent value and the another language-dependent value may be each a string.

[0010AG] The apparatus may further include provisions for setting the locale variable to represent the second locale, and provisions for forming the regenerated query results for presentation for the second locale.

[0010AH] An object may facilitate regenerating the query results by automatically forming the transformed query.

[0010AI] The provisions for determining the key may include provisions for searching a multilingual data source using the language-dependent value and a first locale indicator representing the first locale, and provisions for retrieving the key.

[0010AJ] The provisions for forming the regenerated query may include provisions for searching the multilingual data source using the key and a second locale indicator representing the second locale to retrieve the another language-dependent value.

[0010AK] The provisions for searching the multilingual data source using said key and said second locale indicator may be operable to search without matching the language-dependent value against the data source.

[0010AL] The apparatus may further include provisions for detecting the locale variable is set as the second locale, and provisions for supplying a prompt to a user in a language associated with the second locale to provide a user input including the another language-dependent value. The prompt may facilitate receiving the another language-dependent value dynamically.

[0010AM] The provisions for supplying the prompt may include provisions for retrieving a list of selections in the language from which the another language-dependent value is selected.

[0010AN] Each of the first and the second locales may be unique to a user and the apparatus may further include provisions for specifying a language in which to regenerate query results regardless of geographical location.

[0010AO] In accordance with another aspect of the invention, there is provided an apparatus for implementing multilingual objects to regenerate the results of queries in a language other than one used to initially generate the queries. The apparatus includes provisions for specifying query parameters configured to formulate a query, and provisions for specifying a condition identifying a first translated label value associated with a first language. The apparatus further includes provisions for transforming the condition to form a translated condition as a function of a locale variable, and provisions for implementing the translated condition with the query to form a transformed query that is language-independent prior to generating a report. The transformed query is configured to automatically regenerate the results of the query as part of the report in a second language.

[0010AP] The apparatus may further include provisions for constructing a multilingual object configured to form associations among a base object specifying a number of language-independent keys, a locale field to specify a field in a multilingual data source, and two or more translated label values each of which is associated with a different language.

[0010AQ] The provisions for transforming the condition may further include provisions for using the multilingual object to retrieve a key from the number of language-independent keys that are stored in the multilingual data source, the locale field and the first translated label value may be used to query the multilingual data source.

[0010AR] The provisions for implementing the translated condition may further include provisions for defining the transformed query in terms of the key and a locale variable.

[0010AS] In accordance with another aspect of the invention, there is provided an apparatus for regenerating results of a query in a language other than one used to initially generate the results of the query. The apparatus includes provisions for specifying a language-dependent value of a plurality of language-dependent values for filtering query results in accordance with the language-dependent value, provisions for associating a first locale to the language-dependent value, and provisions for determining a key based on the language-dependent value and on the first locale, the key being common across tables storing information in different languages. The apparatus further includes provisions for generating a transformed query to include a locale variable configured to indicate a second locale for regenerating the query results to form regenerated query results for the second locale using another language-dependent value of the plurality of language-dependent values. The apparatus further includes provisions for detecting the locale variable is set as the second locale, and provisions for supplying a prompt to a user in a language associated with the second locale to provide a user input including the another language-dependent value, wherein the prompt facilitates receiving the another language-dependent value dynamically. The transformed query is applied to the tables to form the regenerated results. The query results are in a first language and the regenerated query results are in a second language, and the language-dependent value and the another language-dependent value are each a string.

[0010AT] The apparatus may further include provisions for setting the locale variable to represent the second locale, and provisions for forming the regenerated query results for presentation for the second locale.

[0010AU] An object may facilitate regenerating the query results by automatically forming the transformed query.

[0010AV] The provisions for determining the key may include provisions for searching a multilingual data source using the language-dependent value and a first locale indicator representing the first locale, and provisions for retrieving the key.

[0010AW] The provisions for forming the regenerated query may include provisions for searching the multilingual data source using the key and a second locale indicator representing the second locale to retrieve the another language-dependent value.

[0010AX] The provisions for searching the multilingual data source using said key and said second locale indicator may be operable to search without matching the language-dependent value against the data source.

[0010AY] The provisions for supplying the prompt may include provisions for retrieving a list of selections in the language from which the another language-dependent value is selected.

[0010AZ] Each of the first and the second locales may be unique to a user and may further include provisions for specifying a language in which to regenerate query results regardless of geographical location.

BRIEF DESCRIPTION OF THE FIGURES

[0011] The invention is more fully appreciated in connection with the following detailed description taken in conjunction with the accompanying drawings, in which:

[0012] FIGs. 1A to 1D illustrate portions of common data structures for conventionally storing and relating data expressed in multiple languages;

[0013] FIG. 2 is a functional block diagram illustrating the formation of a transformed query that preserves conditions as a transformed condition independent of language and/or locale, according to at least one embodiment of the present invention;

[0014] FIG. 3 is a flow diagram exemplifying a method of generating a report using a transformed query, according to an embodiment of the present invention;

[0015] FIG. 4 is a diagram of an apparatus configured to regenerate query results from transformed queries at or for one or more report engines, according to an embodiment of the invention;

[0016] FIGs. 5-11 show examples of the formation of transformed queries using the apparatus of FIG. 4;

[0017] FIG. 12 introduces an architecture in which multilingual semantic layer objects ("MSLOs") are implemented to perform multilingual queries, according to at least one embodiment of the present invention; and

[0018] FIG. 13 shows examples of a base object and a multilingual semantic layer object of FIG. 12, according to one embodiment of the present invention.

[0019] Like reference numerals refer to corresponding parts throughout the several views of the drawings.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0020] FIG. 2 is a functional block diagram 200 illustrating the formation of a transformed query that preserves conditions as transformed conditions, which are independent of language and/or locale, according to at least one embodiment of the present invention. In particular, FIG. 2 depicts a query ("q(0)") 202 being transformed into a transformed query ("q(t)") 230, which is a set of executable instructions defining the information to be used for generating a report in differing languages for different locales. Examples of different locales having different languages are represented as locale ("1") 240, locale ("2") 242, and locale ("N") 244. Note that transformed query 230 is generated, in this example, for a locale specific to computing device 206. Transformed query ("q(t)") 230 includes at least one transformed condition ("c(t)") 220 for preserving an initial condition in a form that is independent of language. Advantageously, the preservation of conditions in accordance with the present invention facilitates filtering of multilingual data sources for

report generation at or for any locale, regardless of language and without manual intervention (e.g., without modify executable instructions to produce a report in another language). By contrast, traditional techniques of filtering multilingual data sources rely on matching word strings input from a user, for example, against a number of fields having various string values, which are typically limited to a specific language. Consequently, if the query is to be conventionally regenerated in another language, then a user is required to manually intervene to, for example, change the selection from one set of fields having various string values in one language to another set of fields specific to another language. This requires specialized knowledge and generally leads to inefficiencies in generating reports in various languages.

[0021] As used herein, the term "query" is used in some embodiments to describe a collection of specifications (or question parameters) used to search for and to extract a set of data requested from a database, such as a multilingual database. Note that the query may be created in a variety of ways, such as being provided manually, through the use of a graphic user interface ("GUI"). Or, it can be programmatically provided based on predetermined conditions. The term "report" generally refers to structured information constituting the results of a query, including titles, page formatting, and the like, in a language specified by preserved conditions, such as preserved semantics. As used herein, the term "condition" is used in some embodiments to describe, in whole or in part, a user input for filtering the results of a query according to certain criteria specified by the user. A condition typically is a string of characters representing a word in a particular language, and in some cases, includes a comparator operator, such as comparative operators "equal to," "greater than," "less than," or the like.

[0022] Transformed condition 220 is a language-independent condition that is used to filter or further refine the results of a query, as specified by condition ("c(o)") 204, but generally in a different language than that initially used to specify condition 204. So by transforming language-dependent condition 204 into a language-independent condition for a first locale, a user associated with a second locale can use transformed condition 220 to run query 202 as transformed query 230, thereby producing a report with similar query results to that expected with condition 204, but in a different form (e.g., in a different language). The preserved condition of transformed condition 220 is used to determine the specific language.

[0023] A computing device 206 generates transformed condition 220 from one or more conditions 204, and does so based on, for example, locality, one or more equivalent translated labels that are each translated into different languages, language-independent keys,

and associations among locality, translated labels and keys. In particular, computing device 206 includes a semantic layer 208 that includes at least business-oriented associations for mapping multilingual queries to data in underlying multilingual databases 205. Included in semantic layer 208, among other things, are multilingual semantic layer objects ("MSLO") 210 each of which at least functions to relate a key to either a locality or a number of translated labels, or both. Computing device 206 also includes locale manager 212 for maintaining and providing information that identifies the locality to specify the language to be used for the query. In general operation, computing device 206 selects an MSLO 210 associated with condition 204 to retrieve a language-independent key from multilingual databases 205. Then, computing device 206 forms transformed condition 220 to include, for example, the retrieved key and a locale variable, which is configured to take on a value provided by a particular locale manager. An exemplary method of forming transformed query 220 is illustrated in FIG. 3.

[0024] A suitable semantic layer 208 for practicing a specific embodiment of the present invention can be implemented using a Business Objects Designer module, which is manufactured by Business Objects SA. As used herein, the term "label" is used in some embodiments to describe a string of one or more characters, such as those characters constituting a description of translated words (e.g., the term "product name" is such a description), and the term "translated label" is used to describe a string representing a translation of a word in one of many languages. Examples of translated labels are "Plane," which is a translated word in English, and "Avion," which is a translated word in French, with both translated labels being associated with the label "product name." As used herein, the term "locale" is used in some embodiments to describe a collection of settings related to, for example, a regional or cultural preference for a particular locale. A locale can define, among other things, a computer environment that determines the date/time display format, default currency units, default measurement system, and user interface language and the language of the contents of generated reports.

[0025] FIG. 3 is a flow diagram exemplifying a method of generating a report using a transformed query, according to an embodiment of the present invention. During the query transformation process 301, a user, for example, specifies one or more translated labels as conditions at 302. The locality, once determined at 304, then predetermines a unique key used to identify the condition, the key being independent of language. An association between the translated label, the locality, and the key, such as that set forth by a multilingual

semantic layer object, is used to locate the key. In particular, when two of the three are known (e.g., locality and translated label), then the third (e.g., the key) can be made known. At 306, the unique key is retrieved. The condition is the then modified into a transformed condition at 308, which includes the retrieved key and a locale variable. Next, a query can be modified at 310 to form a transformed query, which is a language-independent query irrespective of the language used to define the conditions. The transformed query then can be stored for later execution. Or, a user can execute the transformed query during report generating process 303, with the locale variable being satisfied at run-time (i.e., during report generation process 303).

[0026] Consider that a request to generate a report for any locality (or locale) is made at 312 to initiate report generation process 303. If the locality at which the request is being made is the same as the condition set forth at 302, then the query results generated will be in the same language, unless otherwise specified. But if the locality at which the request is being made is different than the locality at which the condition was acquired at 302, then the language of the regenerated report, in whole or in part, will be different than the language used to define the condition. At 314, the specific locality is identified. At 316 a report is generated in a language specific to the locality identified at 314, the locality being included in the transformed condition at run-time.

[0027] FIG. 4 is a diagram of an apparatus configured to regenerate query results from transformed queries at one or more report engines, according to an embodiment of the invention. Apparatus 404 includes a multilingual query engine ("MQE") 414 and a report engine 420, both of which can be implemented in either hardware or software, or in a combination of both. Multilingual query engine 414 implements the associations of multilingual objects to transform queries from one language to another, and report engines 420, 428, and 432 generate reports based on the transformed queries. Multilingual query engine 414 is configured to accept an initial query ("q(o)") 406 and one or more conditions ("c(o)") 408 via user interface 402, for example, to form a transformed query ("q(t)") 416. One or more multilingual data sources 450, such as data source one, or "DS1," "DS2" or "DSn," provide multilingual data from which the query results will be derived. Multilingual query engine 414 accepts the condition in the form of a standard query language, such as SQL. A generalized condition suitable for forming a transformed condition in SQL is:

DATA_SOURCE.MSLO_NAME = "TLV,"

where DATA_SOURCE refers to a multilingual data structure, such as a table, containing the multilingual data of interest and MSLO_NAME refers to the name of the multilingual semantic layer object that is "equal to" the translated label value (i.e., data representing the translated label) submitted by the one performing the query. As an example, consider that query parameters of an initial query include "product names," "revenue" and an indication that a correlation between those two, such as revenue per product, is to be queried. DATA_SOURCE in this example can refer to a table identified as "PRODUCT," which is the data structure containing the product names sought for the query. Further, consider that condition ("c(0)") 408 is specified as an instance of a product name, such as the French product name "Avion," that will be used to filter against multilingual data sources 450.

[0028] Semantic layer process module 412 of multilingual query engine 414 is configured to interact with semantic layer 208 (FIG. 2) and to use associations of multilingual semantic layer objects, such as MSLO 210, to transform condition 408 into a transformed condition ("c(t)") 418 for automatically running a report and/or generate query results in a language suitable for a given locale. An initial step in forming transformed condition 418 is to use the associations of MSLO 410 to determine a language-independent key. In particular, semantic layer process module 412 translates a language-dependent translated label value (e.g., "Avion") into a language-independent key (e.g., "P0010"), with the locale providing for such translation. For example, a locale indicator "FR" can specify that the particular locale is France.

[0029] To determine a key according to a specific embodiment, semantic layer process module 412 employs a generalized SQL statement, such as the following, to transform condition 408 to form a transformed condition 418:

```
SELECT      DATA_SOURCE.KEY
FROM        DATA_SOURCE
WHERE       DATA_SOURCE.MSLO_NAME = 'TLV'
            and
            DATA_SOURCE.LANG = 'XX'
```

where LANG refers to a field in the DATA_SOURCE that will be filtered against the translated label value ("TLV") to match one of the stored translated label values in the data source or table. "XX" represents a locale indicator specifying the language that will be used

to create a report. The locale indicator, XX, can be provided by a locale manager, such as locale manager one ("LM1") 422. Upon matching the input translated label with a translated label value for a specific locale, the associated key can thus be identified.

[0030] Once the semantic layer process module 412 determines the key, then that language-independent key is retrieved and used to form transformed condition 418, which can be expressed in terms of the key and a locale variable. A locale variable is configured to take a value (e.g., locale indicator) at runtime that is indicative of a locale (and/or language) for which a report is generated. To form transformed condition 418 according to a specific embodiment, multilingual query engine 414 defines transformed condition 418 as a generalized SQL statement, such as in the following:

DATA_SOURCE.KEY = '#####' and DATA_SOURCE.LANG = @Language()

where KEY refers to the key identifier '#####' that is language-independent, and @Language() refers to the name of locale variable configured to take the value of a locale indicator for specifying the language that will be used to create a report. So at run-time, the key and the locale (i.e., language) will be known. To produce a transformed query 416, initial query 406 is associated with (or includes) transformed condition 418, whereby transformed query 416 will provide query results in terms of the language used to either run the report or view the report, or both.

[0031] Next, any of report engines 420, 428 or 432 can evaluate transformed query 416 at or for respective locales 1, 2 or n. As an example, consider that report engine 420 is disposed in computing device 404, which resides in France. Further to this example, other report engines, such as report engines 428 and 432, are located in the United States and Germany, respectively. Report engines 428 and 432 receive queries (specifically transformed queries 416) via network 426 for execution. Each report engine includes a locale manager ("LM") for indicating the locale of interest, which in this example refers to a geographic locale. For instance, "LM1" 422 is configured to produce a locale indicator of "FR" for France, whereas "LM2" 430 and "LMn" 434 respectively produce locale indicators of "US" and "DE," for example. In a specific embodiment, report engine 420 includes locale manager ("LMx") 424 for enabling execution (e.g., report generation) at computing device 404 even if the query results are for consumption at another locale. While the preceding example describes the locales as geographic locales, the term "locale" is not restricted to connote just

geographic locations and is intended to have a broader interpretation, as is indicated below with respect to FIG. 12.

[0032] To illustrate how an initial query can be established, consider that multilingual query engine 414 is implemented, in whole or in part, as a module manufactured by Business Objects SA. A user uses multilingual query engine 414 to build a query via a query panel, which is a portion of a user interface that provides business objects. The user then drags and drops objects from the query panel, such as a "Product name" object and a "Revenue" object, into a result pane, which is a functional portion of the user interface that defines the query parameters, thus indicating that the user wants to retrieve correlated values for these two objects (namely, revenue per product). Then, the user drops the "Product Name" object, which in this example is a multilingual semantic layer object, into a filter zone (or condition pane). The filter zone is another functional portion of the user interface defining the conditions of the query. In this case, the conditions indicate that the user wants to select only a certain product name, such as "Avion." At this stage, computing device 404 proposes several comparative operators, such as "equal to_____," "in list _____", "not equal to_____," etc. Consider that the user picks the "equal to" condition and enters a constant, such as the word "Avion." Accordingly, multilingual query engine 414 will query multilingual data sources containing data relating to product names and revenues to generate results for "Avion." FIGs. 5-11 extend the present example to further illustrate the functionalities of multilingual query engine 414, according to at least a specific embodiment of the present invention.

[0033] Further to the foregoing example, consider that FIG. 5 depicts an exemplary multilingual data source 500, according to a specific embodiment of the present invention. In this example, multilingual data source 500 is a table storing multilingual data in accordance with a "duplicative row" schema for relating product identifiers, languages and product names. In particular, column 510 stores product identifiers ("PROD_ID") as keys, column 512 stores language indicators ("LANG") identifying the language to which a product name belongs, and column 514 stores product names ("PROD NAME") as translated labels. FIG. 5 also depicts that multilingual data source 500 is linked to a fact table 550 for storing descriptive information regarding the products of multilingual data source 500. In particular, fact table 550 stores sales facts for the various products, whereby column 560 stores product identifiers and column 562 stores associated sales figures.

[0034] As stated earlier, the user in this example is interested in filtering a query using product name "Avion." As shown in FIG. 6, query ("q(o)") 600 includes generalized structured query language instructions that are configured to cause a multilingual query engine to generate a total sum of a sales, or revenue, for various product names. Query 600 includes a condition 602 for filtering the results of query 600 to yield total sales or revenue for the product named "Avion." Advantageously, condition 602 is easily understood by users forming query 600, such as represented by business objects, and as such, it is evident how query 600 is being filtered. For example, the condition, "Avion," is easily understood by French-speaking users, who should readily identify that query 600 is to be filtered against the term "Avion."

[0035] FIG. 7 depicts generalized structured query language instructions for identifying a key to be used to form a transformed condition. Code snippet 700 is configured to select a product identifier, or key, based on a translated label ("Avion") 702 and a locality associated with a locale indicator ("FR") 704. Upon executing code snippet 700, multilingual query engine will retrieve "P0020" as a key. As shown in FIG. 5, translated label 702 matches at field 522 and locale indicator 704 matches at field 520, thereby identifying field 524 as including key "P0020."

[0036] FIG. 8 depicts a representation of a transformed condition as formed by a multilingual query engine. As shown, transformed condition 800 includes a key ("P0020") 802 and a locale variable ("@Language()") 804. FIG. 9 illustrates formation of a transformed query ("q(t)") 900, according to a specific embodiment of the present invention. In this example, transformed query 900 includes query 600 and transformed condition 902 and represents the semantics of what a user has specified, regardless of their language or locale. In most embodiments, transformed query 900 and/or transformed condition 902 are preserved as part of the query results. So when a French user runs the report, the locale variable will be resolved into the current user's language, such as the locale indicator "FR."

[0037] If a request to run the report is at a French locale, FIG. 10 represents a representative code snippet 1000 as a transformed query for execution for a particular locale. As shown, code snippet 1000 includes transformed condition 1002, with its locale indicator value 1010 being resolved as "FR." As such, FR is used to query a multilingual data source for generating reports in the French language. Also, FIG. 10 includes query results derived from executing transformed query 1000, with the query results being depicted as regenerated report 1040. Note that the original condition that had been input by the user is shown as

translated label "Avion" in field 1050. But if the request to run the report is at a locale in the United States, FIG. 11 represents another code snippet 1100 representing the transformed query for execution at a different locale. As such, code snippet 1100 includes transformed condition 1102 with a locale indicator value ("US") 1110 that has been resolved for generating reports in the English language (and in particular, the English language of the United States). Query results from executing transformed query 1100 are depicted as regenerated report 1140, where the original condition, "Avion," is shown as translated label "Plane" in field 1150 to accommodate English-speaking users in the U.S.

[0038] Note that the examples of FIGs. 5-11 implement a specific type of database schema known as duplicate rows. But various embodiments of the present invention can implement other database schemas, such as "duplicate columns" and "separate translation tables" schemas. First consider that multilingual data source 500 (FIG. 5) includes duplicate columns for product names in different languages, each column being reserved for one language or locale. For instance, one column is reserved for English product names with a title of "Product.EN_NAME," whereas another column is reserved for French product names with a title of "Product.FR_NAME." In this case, associating a unique identifying key value to a translated label and locale indicator can be determined by using the following: WHERE = PRODUCT.@Language()_NAME, where @Language() is a locale variable configured to take a value representing a locale at run-time. Given a condition for filtering multilingual data sources, such as "equal to Avion," the translated label Avion will be matched against fields of a column determined by @Language(). So if users in the locale are French-speaking, @Language() will be resolved to be FR and the column Product.FR_NAME will be queried. Upon finding a match, a unique key can be determined. Thereafter a transformed condition can be formed in a similar fashion described above.

[0039] To determine a key using a duplicate columns schema in accordance with a specific embodiment, semantic layer process module 412 (FIG. 4) employs a generalized SQL statement, such as the following, to transform condition 408 to form a transformed condition 418:

```
SELECT    DATA_SOURCE.KEY
FROM      DATA_SOURCE
WHERE     DATA_SOURCE.@COLUMN(XX,MLSO_NAME) = 'TLV'
```

where @COLUMN refers to a column that includes a field that matches TLV to the MLSO_NAME in the DATA_SOURCE, and "XX" represents a locale indicator specifying the language that will be used to create a report. Note that in this instance, "@COLUMN" represents a set of executable instructions (e.g., a macro) that identifies which one of a number of duplicate columns is used for locating and retrieving the key.

[0040] Next, consider that that multilingual data source 500 includes separate translation tables, each table being reserved for one language or locale. For instance, one table is reserved for English product names with a title of "PRODUCT_EN_US" (and contains product IDs as keys and English product names), and another table is reserved for French product names with a title of "PRODUCT_FR_FR" (and contains product IDs as keys and French product names). In this case, associating a unique identifying key value to a translated label and locale indicator can be determined by using the following: WHERE = PRODUCT_@Language().NAME, where @Language() is a locale variable configured to take a value representing a locale at run-time. Given a user input of "Avion," the translated label Avion will be matched against fields in a table determined by @Language(). If users in the locale are French-speaking, the PRODUCT_FR_FR table will be selected for retrieving a key using French product names. Upon finding a match, a unique key can be determined. Thereafter a transformed condition can be formed.

[0041] To determine a key using a separate translation tables schema in accordance with a specific embodiment, a semantic layer process module employs a generalized SQL statement, such as the following, to transform a condition to form a transformed condition:

```
SELECT    @TABLE(XX,MLSO_NAME).KEY
FROM      @TABLE(XX,MLSO_NAME)
WHERE     @TABLE(XX,MLSO_NAME).MLSO_NAME = 'TLV'
```

where @TABLE refers to a specific table (as a data source) containing a column identified by MLSO_NAME, and "XX" represents a locale indicator specifying the language that will be used to create a report. Note that in this instance, "@TABLE" represents a set of executable instructions (e.g., a macro) that identifies which one of a number of tables (or data sources) is used for identifying and retrieving the key.

[0042] FIG. 12 introduces an architecture in which multilingual semantic layer objects ("MSLOs") are implemented to perform multilingual queries, according to at least

one embodiment of the present invention. In particular, FIG. 12 depicts an exemplary MSLO 1204 being associated to a base object 1202 as well as to its properties represented by, for example, a language field ("LANG") 1206, a key 1208 and one or more translated label ("TLs") 1210. These three properties of MSLO 1204 facilitate the regeneration of query results in any language regardless of the one used to initially generate the query, whereby the regenerated query results require no manual intervention to revise executable program instructions. Although these properties can be used for many purposes, their principal functions are briefly described as follows. LANG 1206 is a field describing a property configured to include a locale variable, the value of which depends on the locale where a report is run or is to be reviewed, thereby indicating the language to be used in regenerating reports. Key 1208 is a property that includes a unique, language-independent identifier for specifying a specific translated label value for a specific value of LANG 1206. In one embodiment, base object ("obj") 1202 provides key 1208 for relating one or more TLs 1210 having a synonymous meaning. TLs 1210 is a field describing a property configured to contain one of a number of translated label values, such as "Avion," when LANG 1206 indicates a locale indicator, "FR," to indicate a French-speaking locale.

[0043] FIG. 13 shows examples of a base object 1202 and an MSLO 1204, according to one embodiment. In FIG. 13, base object 1310 is associated with a name indicative of its function—it is associated with a product identifier, or "ID." Here, its name is maintained as property 1312. Similarly, MSLO 1330 is associated with a name for specifying a particular label and is described by property 1332. Here, the particular label to be translated into any one language represents a "product name." Further, MSLO 1330 includes a MSLO property 1334 indicating whether object 1330 is a multilingual object. In this example, MSLO 1330 includes an affirmative indication, "Y," as property 1334 to indicate that object 1330 is an MSLO. MSLO 1330 can also include a product ID (or a reference), such as P0020, as the unique language-independent key in property 1336. Further, MSLO 1330 includes a property 1338 for maintaining a locale variable for representing the language with which to query multilingual data sources.

[0044] Referring back to FIG. 12, both base object 1202 and MSLO 1204 are disposed within a semantic layer 1224 of a computing device 1220. Computing device 1220 includes a number of standard components, including a CPU 1240, input/output ("I/O") devices 1242, such as a graphical user interface, and a network interface ("I/F") 1244, all of which communicate via a bus 1246. Network I/F 1244 facilitates communications via any

kind of network (not shown) to another computing device 1220 (not shown) for another locale, or to a number of various multilingual data sources 1250, each of which are represented as data source one ("DS1"), data source two ("DS2"), and data source "n" ("DSn").

[0045] Computing device 1220 also includes a memory 1218 that is also connected to bus 1246. Memory 1218 stores a set of executable programs that are used to implement the functions of the various embodiments of the present invention. In one embodiment of the invention, subsets of executable program instructions constitute a program engine 1221, a multilingual query engine ("MQE") 1230, a report engine 1232 and a locale manager 1234. Multilingual query engine 1230, report engine 1232 and locale manager 1234 have similar structures and/or functions as described above. Program engine 1221 includes various layers representing abstractions of program instructions, including upper layer 1222, semantic layer 1224 and data source interface layer 1226. Upper layer 1222 includes higher-level data representations and/or executable program instructions for effectuating multilingual querying, such as providing a graphical user interface (e.g., for accepting query data and for presenting report data) as well as application layer processing. Semantic layer 1224 includes intermediate-level data representations and/or executable program instructions for managing MSLOs 1204 and other business-related objects, according to one embodiment of the present invention. Objects such as MSLOs 1204 can be referred to as semantic layer objects as they are used to translate relatively sophisticated database code into business objects in terms that are easily understood by those having little or no experience in database programming. Data source interface layer 1226 includes lower-level data representations and/or executable program instructions for managing the exchange of data among computing device 1220 and multilingual data sources 1250.

[0046] As indicated above, the term locale is intended to have a broad meaning that describes more than a particular physical location or place. In various embodiments of the present invention, the term "locale" can also refer to a collection of settings defining a computer environment. A computer environment can be unique to a specific person (i.e., a unique user ID), a specific computing device, a specific applications program with which to generate a multilingual report, and the like. Consider that computing device 1220 is a server that is configured, for example, as computing device 206 (FIG. 2) for communicating with different locales, such as locale ("1") 240, locale ("2") 242, and locale ("N") 244 (all of FIG. 2). Each of locals 240-244 can represent a user and/or a computing device sharing a common

network with computing device 1220. In the case that each locale is associated with a unique user, then when a user logs into a networked computing device (or is otherwise identified by computing device 1220), locale manager 1234 establishes a specific computer environment for that user, for example, by applying a collection of settings that determine a specific language for the user interface. Therefore, two or more users that are conversant in different languages can share a common user interface or computing device that uses report engine 1232 to generate language-specific reports based on a transformed query running. So when a user requests a report, locale manager 1234 uses a locale indicator, such as a user identifier ("ID"), to format the report in a language that is suitable for that user. For example, a locale indicator ("JOHNDOE123") can identify a user named John Doe and can associate that person to particular computing environment using a preferred language, such as English.

[0047] It should be appreciated that the executable modules illustrated in memory 1218 are exemplary. The functions of the invention may be implemented in any number of ways. In addition, it should be appreciated that functions of the invention need not be implemented on a single computing device 1220. The functions of the various embodiments of the present invention may be implemented across a set of servers, clients, clients and servers, etc. It is the functions of various embodiments that are significant, not where or how these functions are implemented.

[0048] In at least one embodiment of the present invention, a user can dynamically specify a translated label as a condition to filter multilingual databases rather than statically specifying that label. In particular, the following condition can be referred to as a static condition: PRODUCT.PROD_NAME = "Avion." An example of procuring a condition dynamically can be expressed as:

PRODUCT.PROD_NAME = @prompt("choose a product",PRODUCT_Name),

where a user running the query will typically be presented with a dialog box called "choose a product." The user can then be presented with a list of possible values for the various product names in a multilingual database. When the user selects a particular product name to filter, the selected value then replaces the '@prompt' string in the query so that it if the user picks, for example, the product name "bicyclette" in the French language, then only query results for this product are reported. When the report is rerun at another locale having a different language, a list of values is presented to a user with those values being product names in a

language determined by locale variable @Language(). Given the selected product name, as a translated label, and specific locale indicator, a transformed condition can be determined in a manner similarly described above.

[0049] Although some of the foregoing examples describe using an initial query in one language for regenerating query results to form a report in another language, the various embodiments of the present invention are not limited to those examples. In at least one embodiment, a report being run at a particular locale might be composed of multiple subqueries, at least two of which have been formed in different languages, or at or for different locales. The two subqueries can be formulated in languages that are unlike that used to run the report. In alternative embodiments, "nested" subqueries can be used to form regenerated query results to create a report. For example, regenerate query results in one language, which are based on an initial query in another language, can be embedded in or form part of another query that yields regenerated results in yet another language. An ordinarily skilled artisan will appreciate that the embodiments of the present invention can be implemented in a variety of ways.

[0050] An embodiment of the present invention relates to a computer storage product with a computer-readable medium having computer code thereon for performing various computer-implemented operations. The media and computer code may be those specially designed and constructed for the purposes of the present invention, or they may be of the kind well known and available to those having skill in the computer software arts. Examples of computer-readable media include, but are not limited to: magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD-ROMs and holographic devices; magneto-optical media such as floptical disks; and hardware devices that are specially configured to store and execute program code, such as application-specific integrated circuits ("ASICs"), programmable logic devices ("PLDs") and ROM and RAM devices. Examples of computer code include machine code, such as produced by a compiler, and files containing higher-level code that are executed by a computer using an interpreter. For example, an embodiment of the invention may be implemented using Java, C++, or other object-oriented programming language and development tools. Another embodiment of the invention may be implemented in hardwired circuitry in place of, or in combination with, machine-executable software instructions.

[0051] The foregoing description, for purposes of explanation, used specific nomenclature to provide a thorough understanding of the invention. However, it will be

apparent to one skilled in the art that specific details are not required in order to practice the invention. Thus, the foregoing descriptions of specific embodiments of the invention are presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed; obviously, many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, they thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the following claims and their equivalents define the scope of the invention.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A method of regenerating results of a query in a language other than one used to initially generate the results of the query, the method comprising:

specifying a language-dependent value of a plurality of language-dependent values for filtering query results in accordance with said language-dependent value;

associating a first locale to said language-dependent value;

determining a key based on said language-dependent value and on said first locale, said key being language-independent; and

generating a transformed query to include the determined key and a locale variable configured to indicate a second locale for regenerating said query results to form regenerated query results for said second locale using another language-dependent value of said plurality of language-dependent values.

2. The method of claim **1** wherein said query results are in a first language and said regenerated query results are in a second language, and said language-dependent value and said another language-dependent value are each a string.

3. The method of any one of claims **1** and **2** further comprising:

setting said locale variable to represent said second locale; and

forming said regenerated query results for presentation for said second locale.

4. The method of any one of claims **1** to **3** wherein an object facilitates regenerating said query results by automatically forming said transformed query.

5. The method of claim **3** wherein determining said key comprises:

searching a multilingual data source using said language-dependent value and a first locale indicator representing said first locale; and

retrieving said key.

6. The method of claim **5** wherein forming said regenerated query comprises searching said multilingual data source using said key and a second locale indicator representing said second locale to retrieve said another language-dependent value.

7. The method of claim **6** wherein searching said multilingual data source using said key and said second locale indicator is done without matching said language-dependent value against said data source.

8. The method of any one of claims **1** to **7** further comprising:

detecting said locale variable is set as said second locale; and

supplying a prompt to a user in a language associated with said second locale to provide a user input including said another language-dependent value,

wherein said prompt facilitates receiving said another language-dependent value dynamically.

9. The method of claim **8** wherein supplying said prompt comprises retrieving a list of selections in said language from which said another language-dependent value is selected.

10. The method of any one of claims **1** to **9** wherein each of said first and said second locales is unique to a user and further comprising specifying a language in which to regenerate query results regardless of geographical location.

11. A method of implementing multilingual objects to regenerate the results of queries in a language other than one used to initially generate the queries, the method comprising:

specifying query parameters configured to formulate a query;

specifying a condition identifying a first translated label value associated with a first language;

transforming said condition to form a translated condition as a function of a locale variable; and

implementing said translated condition with said query to form a transformed query that is language-independent prior to generating a report,

wherein said transformed query is configured to automatically regenerate the results of said query as part of said report in a second language.

12. The method of claim **11** further comprising constructing a multilingual object configured to form associations among

a base object specifying a number of language-independent keys,

a locale field to specify a field in a multilingual data source, and

two or more translated label values each of which is associated with a different language.

13. The method of claim **12** wherein transforming said condition further comprises using said multilingual object to retrieve a key from said number of language-independent keys that are stored in said multilingual data source, said locale field and said first translated label value being used to query said multilingual data source.

14. The method of claim **13** wherein implementing said translated condition further comprises defining said transformed query in terms of said key and a locale variable.

15. An apparatus for preserving filtering conditions to regenerate query results using multilingual data sources for various locales, said apparatus comprising:

a locale manager configured to provide at least a locale indicator identifying a first language; and

a multilingual query engine configured to access data sources each maintaining data in multiple languages, and configured further to accept query parameters of a query, acquire a condition for filtering said query to provide query results in terms of said condition, and transform said query into a transformed query, wherein said transformed query is configured to provide said query results in any language as determined by one of a plurality of locale indicators.

16. The apparatus of claim **15** further comprising a report engine configured to generate a report including said query results in a language specified by said one of said plurality of locale indicators.

17. The apparatus of any one of claims **15** and **16** further comprising a data store for maintaining multilingual objects, at least one of which defines associations among said condition, any of said plurality of locale indicators, and at least a language-independent key.

18. The apparatus of claim **17** wherein said multilingual query engine is further configured to retrieve said key from said data sources using said locale indicator and said condition.

19. The apparatus of any one of claims **17** and **18** wherein said multilingual query engine is further configured to form said transformed query by associating said query with said key and a variable capable of having a value equivalent to any of said plurality of locale indicators.

20. A method of regenerating results of a query in a language other than one used to initially generate the results of the query, the method comprising:

specifying a language-dependent value of a plurality of language-dependent values for filtering query results in accordance with said language-dependent value;

associating a first locale to said language-dependent value;

determining a key based on said language-dependent value and on said first locale, said key being common across tables storing information in different languages;

generating a transformed query to include a locale variable configured to indicate a second locale for regenerating said query results to form regenerated query results for said second locale using another language-dependent value of said plurality of language-dependent values;

detecting said locale variable is set as said second locale; and

supplying a prompt to a user in a language associated with said second locale to provide a user input including said another language-dependent value, wherein said prompt facilitates receiving said another language-dependent value dynamically;

wherein said transformed query is applied to said tables to form said regenerated results;

wherein said query results are in a first language and said regenerated query results are in a second language, and said language-dependent value and said another language-dependent value are each a string.

21. The method of claim **20** further comprising:

setting said locale variable to represent said second locale; and

forming said regenerated query results for presentation for said second locale.

22. The method of any one of claims **20** and **21** wherein an object facilitates regenerating said query results by automatically forming said transformed query.

23. The method of claim **21** wherein determining said key comprises:

searching a multilingual data source using said language-dependent value and a first locale indicator representing said first locale; and

retrieving said key.

24. The method of claim **23** wherein forming said regenerated query comprises searching said multilingual data source using said key and a second locale indicator representing said second locale to retrieve said another language-dependent value.

25. The method of claim **24** wherein searching said multilingual data source using said key and said second locale indicator is done without matching said language-dependent value against said data source.

26. The method of any one of claims **20** to **25** wherein supplying said prompt comprises retrieving a list of selections in said language from which said another language-dependent value is selected.

27. The method of any one of claims **20** to **26** wherein each of said first and said second locales is unique to a user and further comprising specifying a language in which to regenerate query results regardless of geographical location.

28. A computer system for preserving filtering conditions to regenerate query results using multilingual data sources for various locales, said computer system comprising;

a processor and a memory;

a locale manager configured to provide at least a locale indicator identifying a first language;

a multilingual query engine configured to access multilingual data sources, and configured further to accept query parameters of a query, acquire a condition for filtering said query to provide query results in terms of said condition, and transform said query into a transformed query for application to said multilingual data sources;

a data store for maintaining multilingual objects, at least one of which defines associations among said condition, any of said plurality of locale indicators, and at least a language-independent key;

wherein said transformed query is configured to provide said query results in any language as determined by one of a plurality of locale indicators;

wherein said multilingual query engine is further configured to retrieve said key from said data sources using said locale indicator and said condition;

wherein said multilingual query engine is further configured to form said transformed query by associating said query with said key and a variable capable of having a value equivalent to any of said plurality of locale indicators.

29. The computer system of claim **28** further comprising a report engine configured to generate a report including said query results in a language specified by said one of said plurality of locale indicators.

30. A computer-readable medium having stored thereon computer code for directing a computer to execute the method of any one of claims **1** to **14** and **20** to **27**.

31. An apparatus for regenerating results of a query in a language other than one used to initially generate the results of the query, the apparatus comprising:

means for specifying a language-dependent value of a plurality of language-dependent values for filtering query results in accordance with said language-dependent value;

means for associating a first locale to said language-dependent value;

means for determining a key based on said language-dependent value and on said first locale, said key being language-independent; and

means for generating a transformed query to include the determined key and a locale variable configured to indicate a second locale for regenerating said query results to form regenerated query results for said second locale using another language-dependent value of said plurality of language-dependent values.

32. The apparatus of claim **31** wherein said query results are in a first language and said regenerated query results are in a second language, and said language-dependent value and said another language-dependent value are each a string.

33. The apparatus of any one of claims **31** and **32** further comprising:

means for setting said locale variable to represent said second locale; and

means for forming said regenerated query results for presentation for said second locale.

34. The apparatus of any one of claims **31** to **33** wherein an object facilitates regenerating said query results by automatically forming said transformed query.

35. The apparatus of claim **33** wherein said means for determining said key comprises:

means for searching a multilingual data source using said language-dependent value and a first locale indicator representing said first locale; and

means for retrieving said key.

36. The apparatus of claim **35** wherein said means for forming said regenerated query comprises means for searching said multilingual data source using said key and a

second locale indicator representing said second locale to retrieve said another language-dependent value.

37. The apparatus of claim **36** wherein said means for searching said multilingual data source using said key and said second locale indicator is operable to search without matching said language-dependent value against said data source.

38. The apparatus of any one of claims **31** to **37** further comprising:

means for detecting said locale variable is set as said second locale; and

means for supplying a prompt to a user in a language associated with said second locale to provide a user input including said another language-dependent value,

wherein said prompt facilitates receiving said another language-dependent value dynamically.

39. The apparatus of claim **38** wherein said means for supplying said prompt comprises means for retrieving a list of selections in said language from which said another language-dependent value is selected.

40. The apparatus of any one of claims **31** to **39** wherein each of said first and said second locales is unique to a user and further comprising means for specifying a language in which to regenerate query results regardless of geographical location.

41. An apparatus for implementing multilingual objects to regenerate the results of queries in a language other than one used to initially generate the queries, the apparatus comprising:

means for specifying query parameters configured to formulate a query;

means for specifying a condition identifying a first translated label value associated with a first language;

means for transforming said condition to form a translated condition as a function of a locale variable; and

means for implementing said translated condition with said query to form a transformed query that is language-independent prior to generating a report,

wherein said transformed query is configured to automatically regenerate the results of said query as part of said report in a second language.

42. The apparatus of claim **41** further comprising means for constructing a multilingual object configured to form associations among

a base object specifying a number of language-independent keys,

a locale field to specify a field in a multilingual data source, and

two or more translated label values each of which is associated with a different language.

43. The apparatus of claim **42** wherein said means for transforming said condition further comprises means for using said multilingual object to retrieve a key from said number of language-independent keys that are stored in said multilingual data source, said locale field and said first translated label value being used to query said multilingual data source.

44. The apparatus of claim **43** wherein said means for implementing said translated condition further comprises means for defining said transformed query in terms of said key and a locale variable.

45. An apparatus for regenerating results of a query in a language other than one used to initially generate the results of the query, the apparatus comprising:

means for specifying a language-dependent value of a plurality of language-dependent values for filtering query results in accordance with said language-dependent value;

means for associating a first locale to said language-dependent value;

means for determining a key based on said language-dependent value and on said first locale, said key being common across tables storing information in different languages;

means for generating a transformed query to include a locale variable configured to indicate a second locale for regenerating said query results to form regenerated query results for said second locale using another language-dependent value of said plurality of language-dependent values;

means for detecting said locale variable is set as said second locale; and

means for supplying a prompt to a user in a language associated with said second locale to provide a user input including said another language-dependent value, wherein said prompt facilitates receiving said another language-dependent value dynamically;

wherein said transformed query is applied to said tables to form said regenerated results;

wherein said query results are in a first language and said regenerated query results are in a second language, and said language-dependent value and said another language-dependent value are each a string.

46. The apparatus of claim **45** further comprising:

means for setting said locale variable to represent said second locale; and

means for forming said regenerated query results for presentation for said second locale.

47. The apparatus of any one of claims **45** and **46** wherein an object facilitates regenerating said query results by automatically forming said transformed query.

48. The apparatus of claim **46** wherein said means for determining said key comprises:

means for searching a multilingual data source using said language-dependent value and a first locale indicator representing said first locale; and

means for retrieving said key.

49. The apparatus of claim **48** wherein said means for forming said regenerated query comprises means for searching said multilingual data source using said key and a second locale indicator representing said second locale to retrieve said another language-dependent value.

50. The apparatus of claim **49** wherein said means for searching said multilingual data source using said key and said second locale indicator is operable to search without matching said language-dependent value against said data source.

51. The apparatus of any one of claims **45** to **50** wherein said means for supplying said prompt comprises means for retrieving a list of selections in said language from which said another language-dependent value is selected.

52. The apparatus of any one of claims **45** to **51** wherein each of said first and said second locales is unique to a user and further comprising means for specifying a language in which to regenerate query results regardless of geographical location.

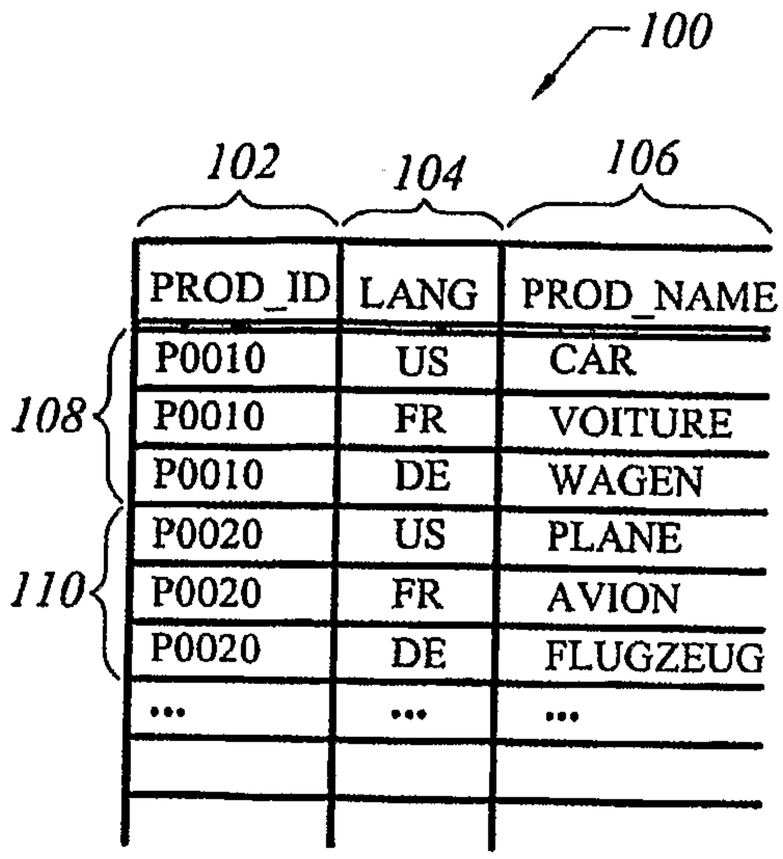


FIG. 1A
(Prior Art)

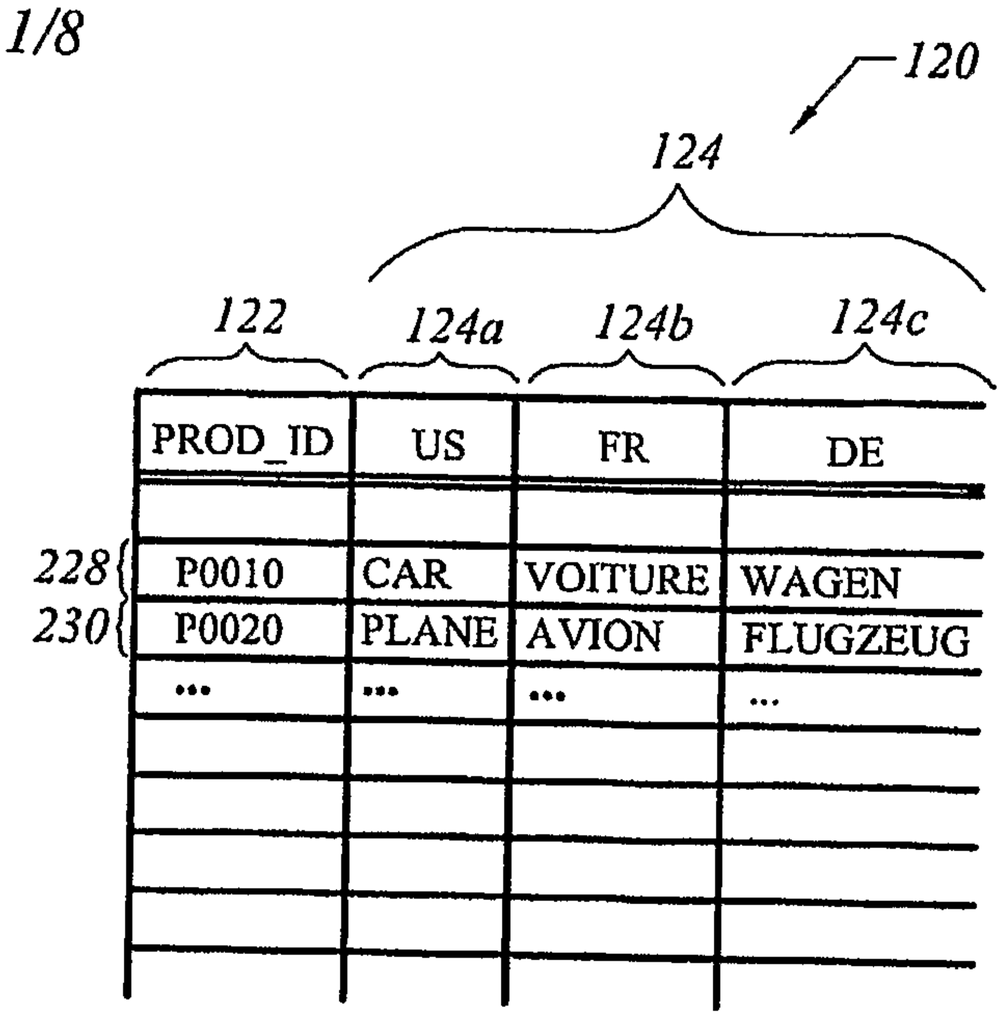


FIG. 1B
(Prior Art)

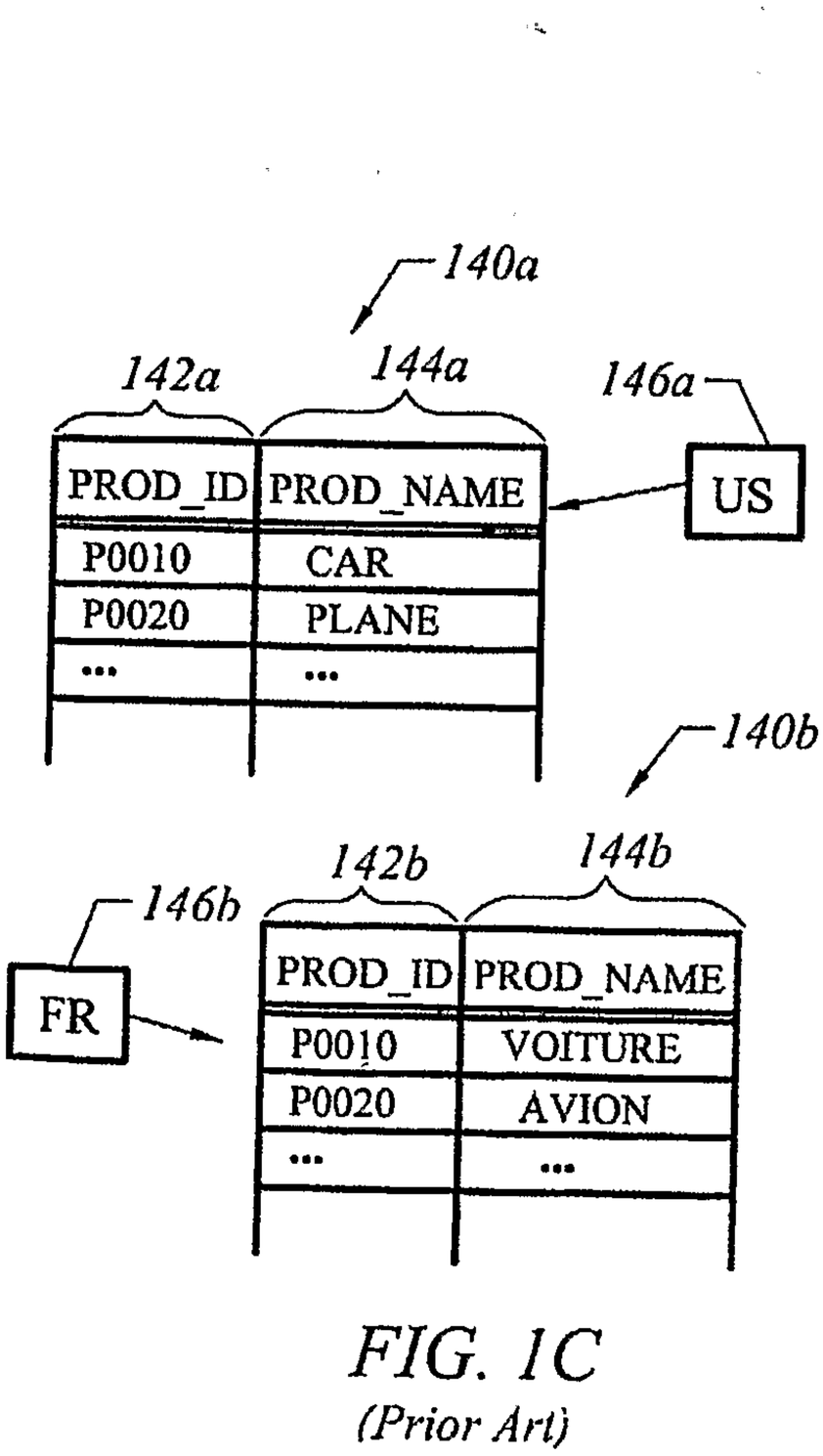


FIG. 1C
(Prior Art)

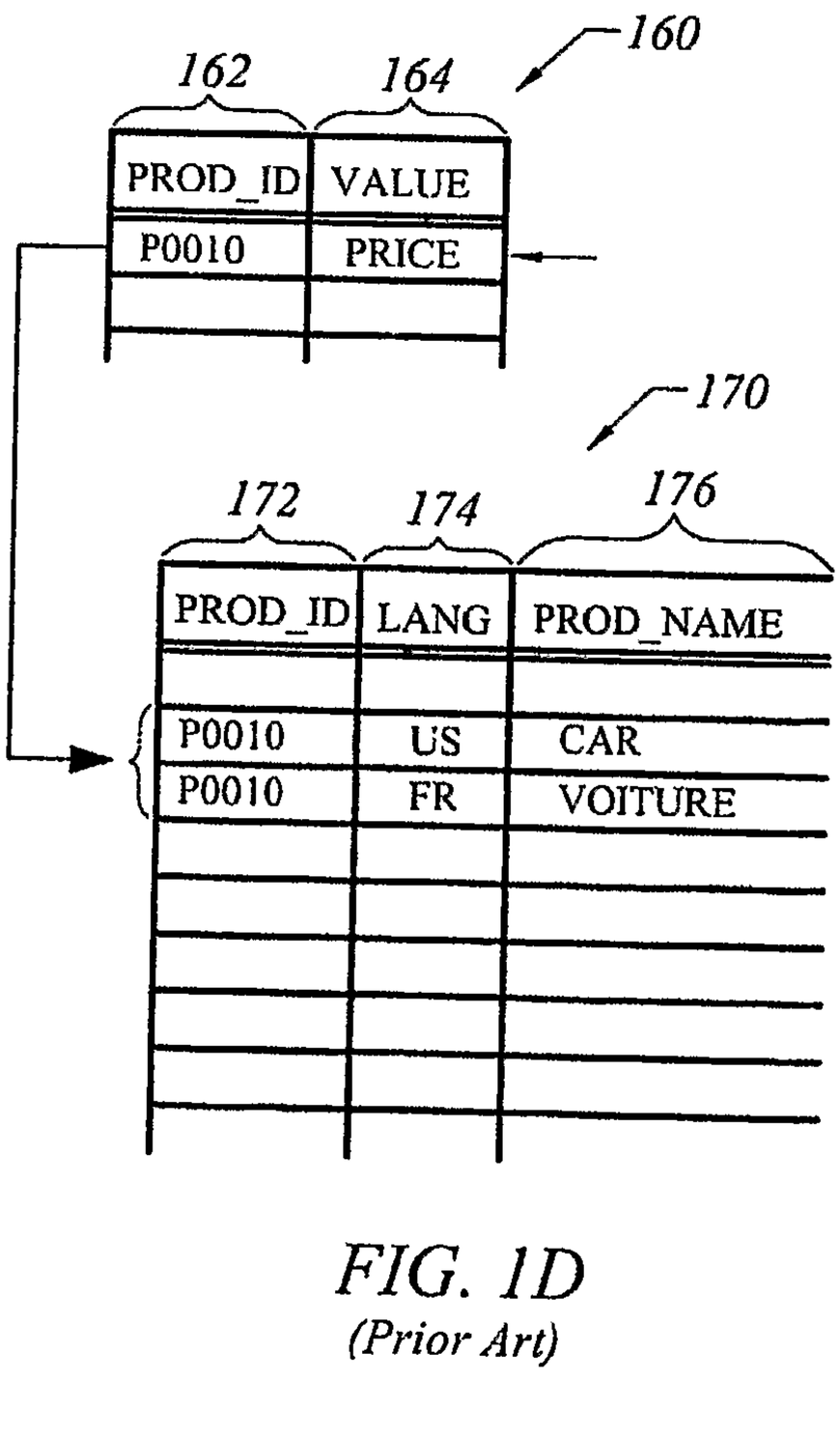


FIG. 1D
(Prior Art)

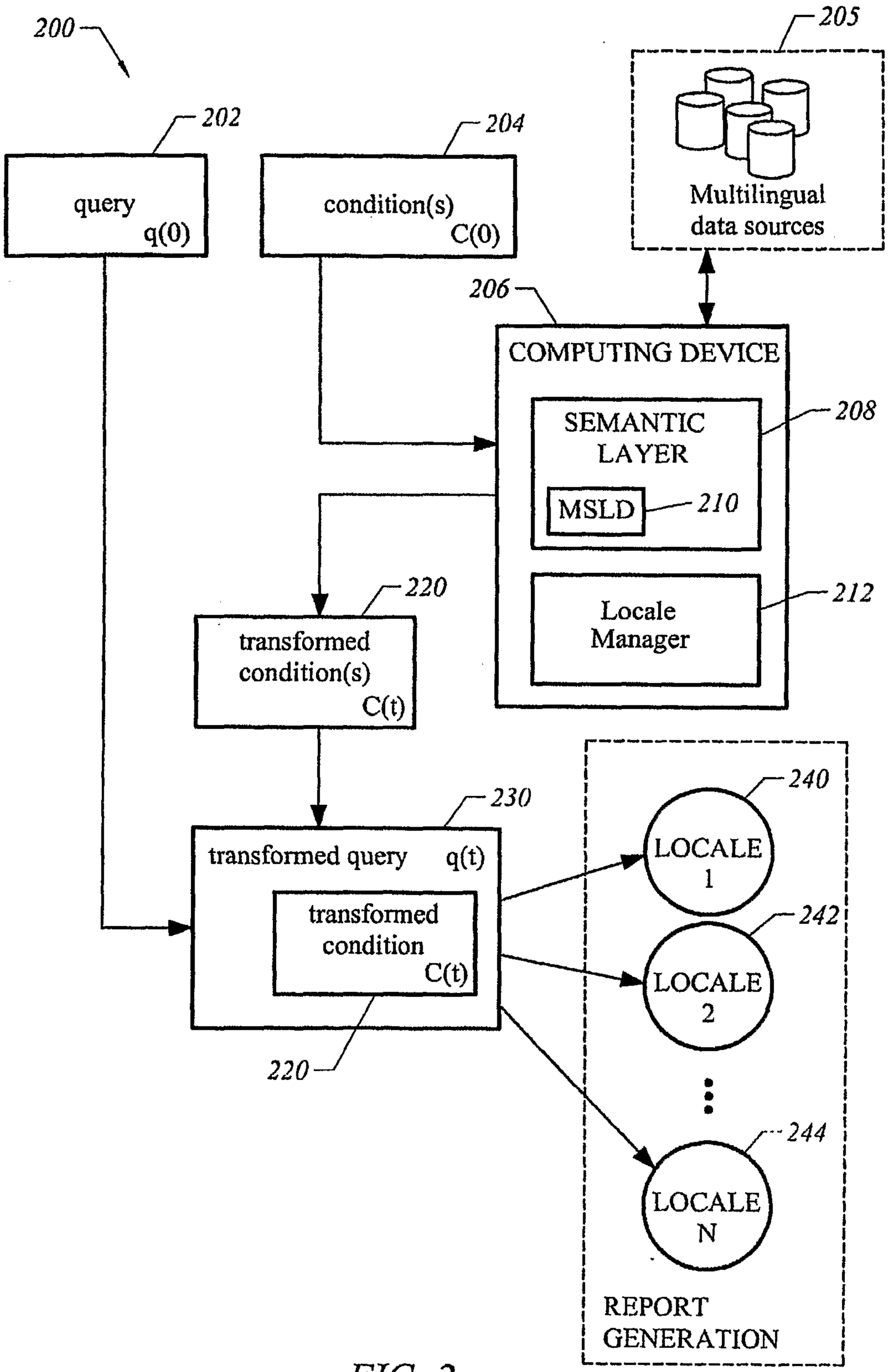

200
202
query
q(0)
204
condition(s)
C(0)
205
Multilingual
data sources
206
COMPUTING DEVICE
SEMANTIC
LAYER
208
MSLD
210
Locale
Manager
212
220
transformed
condition(s)
C(t)
230
transformed query q(t)
transformed
condition
C(t)
220
240
LOCALE
1
242
LOCALE
2
244
LOCALE
N
REPORT
GENERATION

FIG. 2

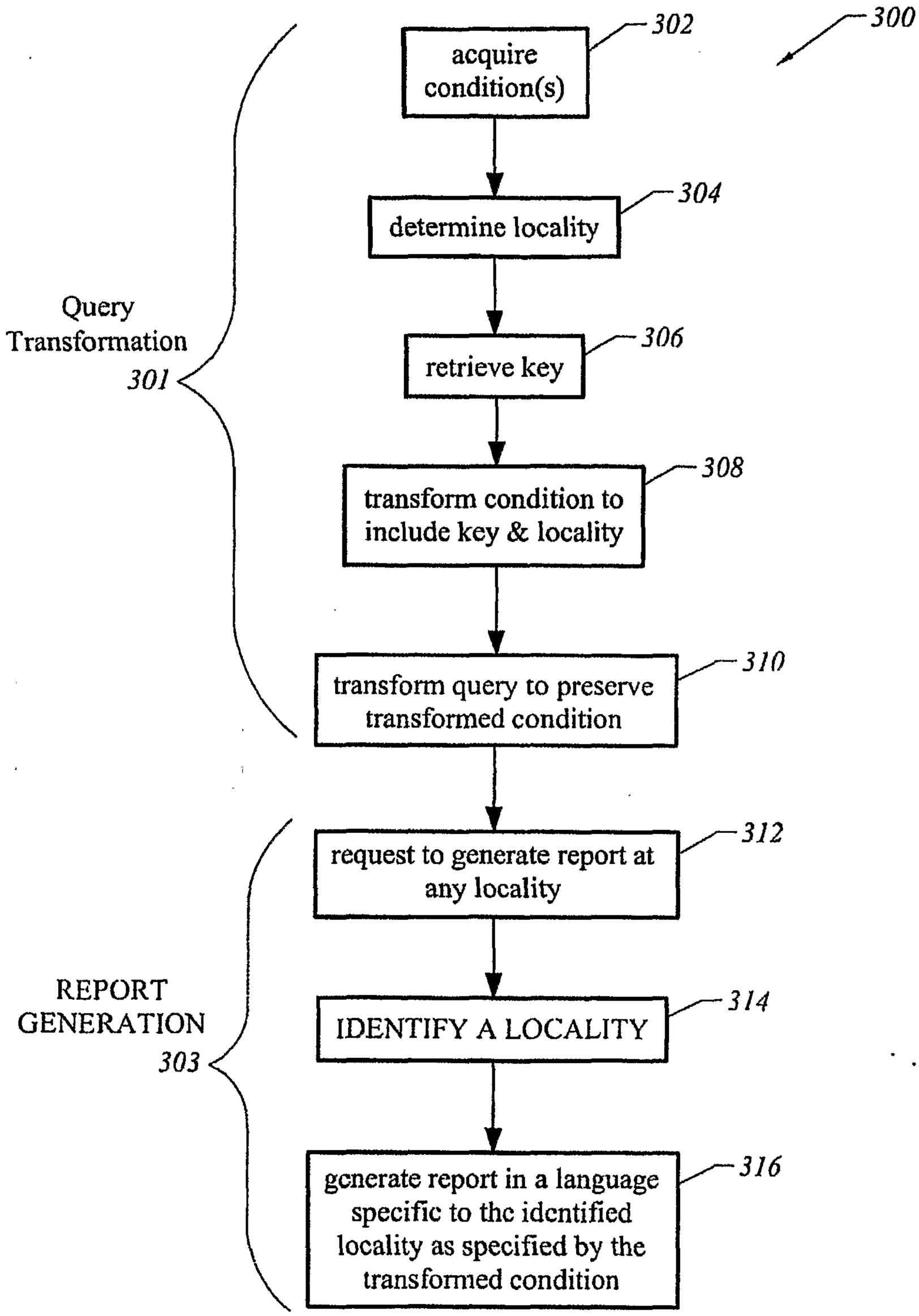

300
acquire condition(s)
302
determine locality
304
retrieve key
306
transform condition to include key & locality
308
transform query to preserve transformed condition
310
request to generate report at any locality
312
IDENTIFY A LOCALITY
314
generate report in a language specific to the identified locality as specified by the transformed condition
316
Query Transformation
301
REPORT GENERATION
303

FIG. 3

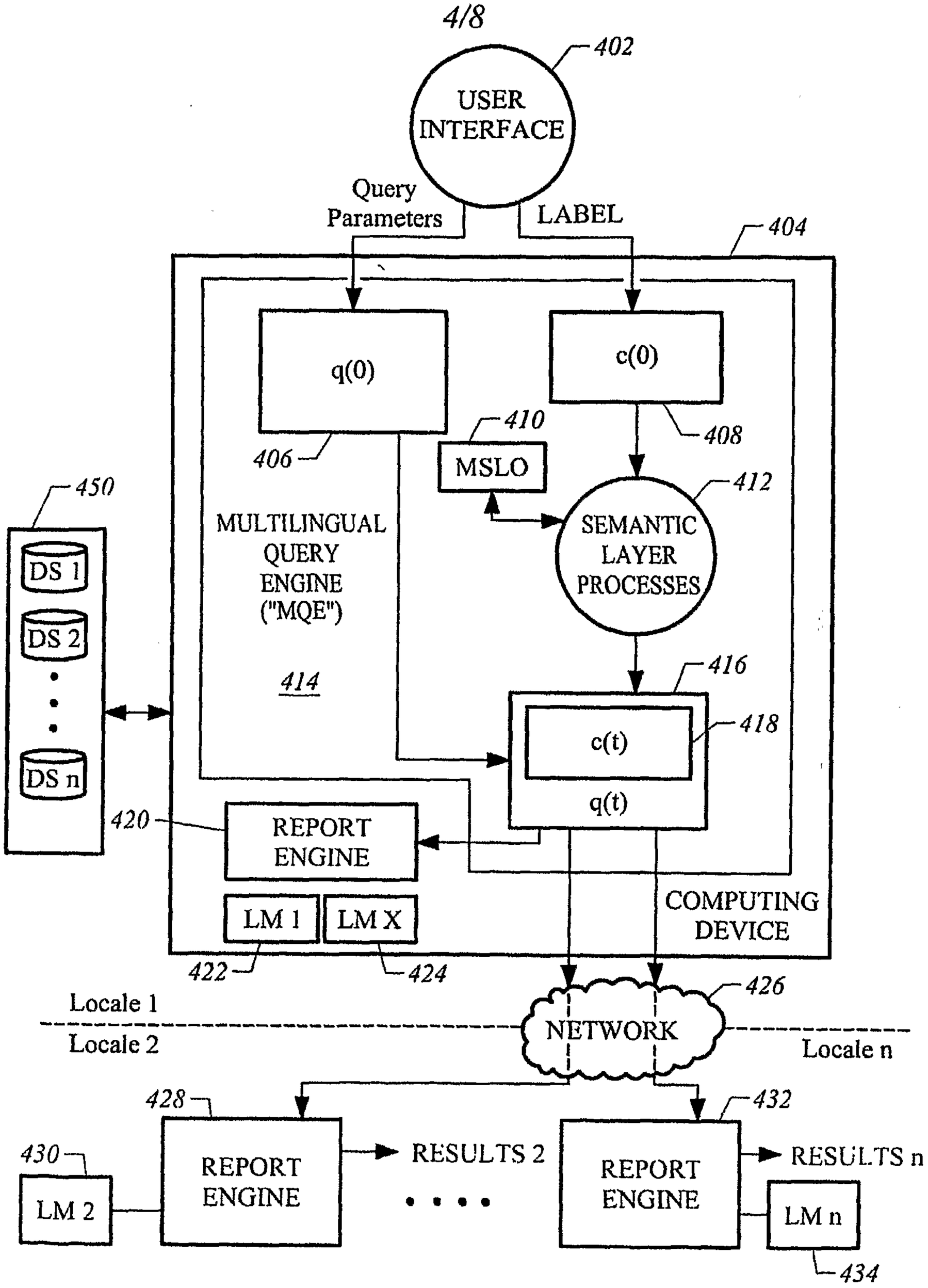

USER INTERFACE
402
Query Parameters
LABEL
404
q(0)
c(0)
406
410
MSLO
408
412
MULTILINGUAL QUERY ENGINE ("MQE")
414
SEMANTIC LAYER PROCESSES
416
450
DS 1
DS 2
DS n
c(t)
418
q(t)
420
REPORT ENGINE
LM 1
LM X
422
424
COMPUTING DEVICE
426
Locale 1
Locale 2
NETWORK
Locale n
428
432
430
LM 2
REPORT ENGINE
RESULTS 2
REPORT ENGINE
RESULTS n
LM n
434

FIG. 4

'PRODUCT' TABLE

PROD_ID	LANG	PROD NAME
P0010	US	Car
P0010	FR	Voiture
P0010	DE	Wagen
P0020	US	Plane
P0020	FR	Avion
P0020	DE	Flugzeug
...	...	...

'FACT' TABLE

PROD_ID	SALES
P0010	2,473
P0020	5,203
...	...

FIG. 5

```
SELECT      PRODUCT.PROD_ID, PRODUCT.PROD_NAME, SUM(FACT.SALES)
FROM        PRODUCT, FACT
WHERE       PRODUCT.PROD_NAME= 'Avion'
            AND
            PRODUCT.PROD_ID=FACT.PROD_ID
GROUPBY     PRODUCT.PROD_ID
```

FIG. 6

```
SELECT      PRODUCT.PROD ID
FROM        PRODUCT
WHERE       PRODUCT.PROD NAME= 'Avion'
            AND
            PRODUCT.LANG= 'FR'
```

FIG. 7

```
PRODUCT.PROD_ID= "P0020" AND PRODUCT.LANG= @Language()
```

FIG. 8

```
SELECT    PRODUCT.PROD_ID, PRODUCT.PROD_NAME, SUM(FACT.SALES)
FROM      PRODUCT, FACT
WHERE     PRODUCT.PROD_ID = 'P0020'          C(t)
          AND
          PRODUCT.LANG= @Language()
          AND
          PRODUCT.PROD_ID=FACT.PROD_ID
GROUPBY   PRODUCT.PROD_ID                    q(t)
```

900, 600, 902

FIG. 9

```
SELECT    PRODUCT.PROD ID, PRODUCT.PROD_NAME, SUM(FACT.SALES)
FROM      PRODUCT, FACT
WHERE     PRODUCT.PROD ID = 'P0020'
          AND
          PRODUCT.LANG = 'FR'
          AND
          PRODUCT.PROD ID = FACT.PROD ID
GROUPBY   PRODUCT.PROD_ID                    q(t@FR)
```

1000, 1002, 1010

Product Name	Revenue
Avion	5,203

1040, 1050

FIG. 10

```
SELECT    PRODUCT.PROD_ID, PRODUCT.PROD NAME, SUM(FACT.SALES)
FROM      PRODUCT, FACT
WHERE     PRODUCT.PROD ID = 'P0020'
          AND
          PRODUCT.LANG= 'US'
          AND
          PRODUCT.PROD_ID=FACT.PROD_ID
GROUPBY   PRODUCT.PROD ID                    q(t@US)
```

1100, 1102, 1110

Product Name	Revenue
Plane	5,203

1140, 1150

FIG. 11

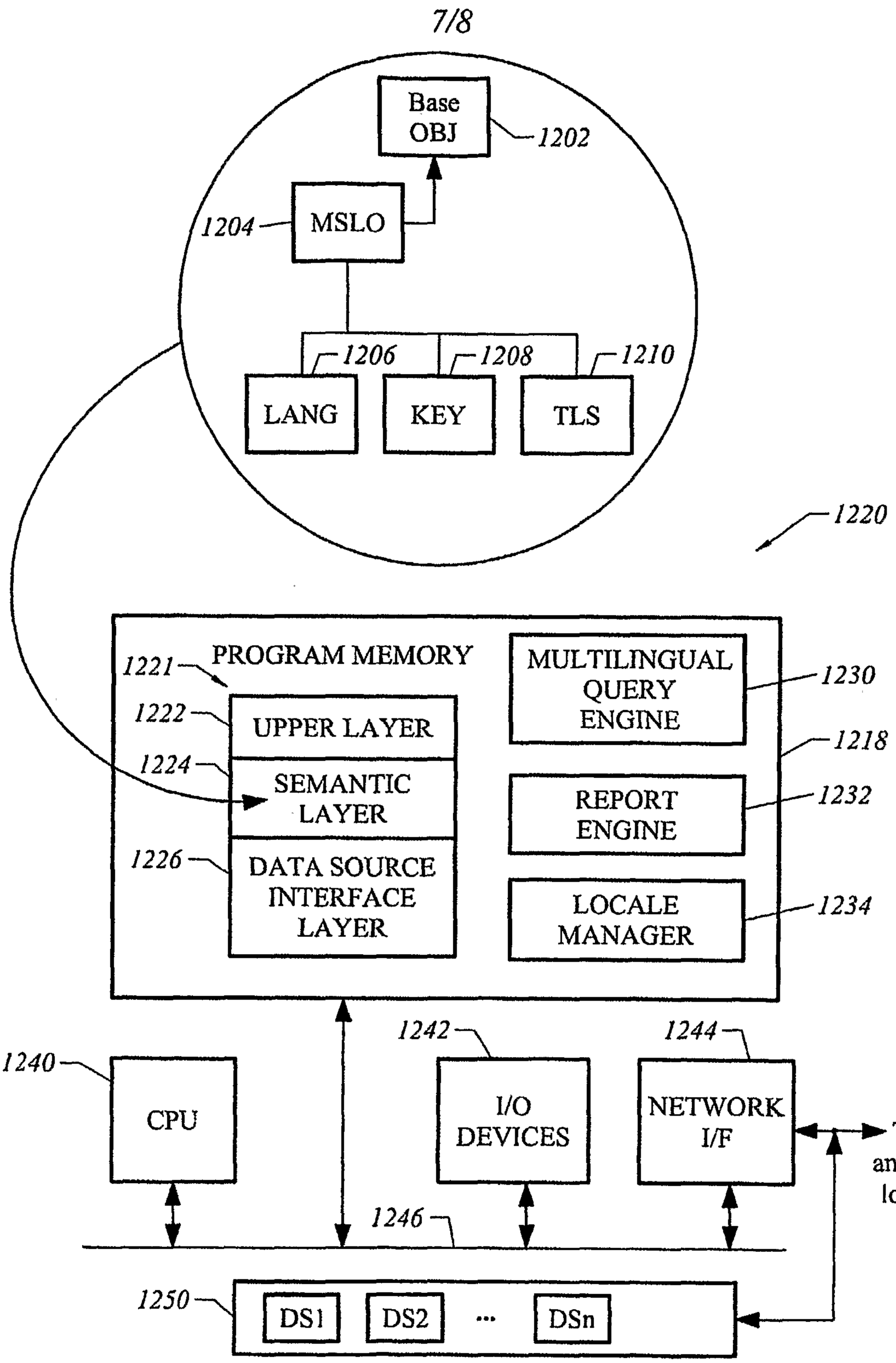

Base OBJ
1202
1204
MSLO
1206
1208
1210
LANG
KEY
TLS
1220
1221
PROGRAM MEMORY
1222
UPPER LAYER
1224
SEMANTIC LAYER
1226
DATA SOURCE INTERFACE LAYER
MULTILINGUAL QUERY ENGINE
1230
1218
REPORT ENGINE
1232
LOCALE MANAGER
1234
1240
CPU
1242
I/O DEVICES
1244
NETWORK I/F
To another locale
1246
1250
DS1
DS2
...
DSn

FIG. 12

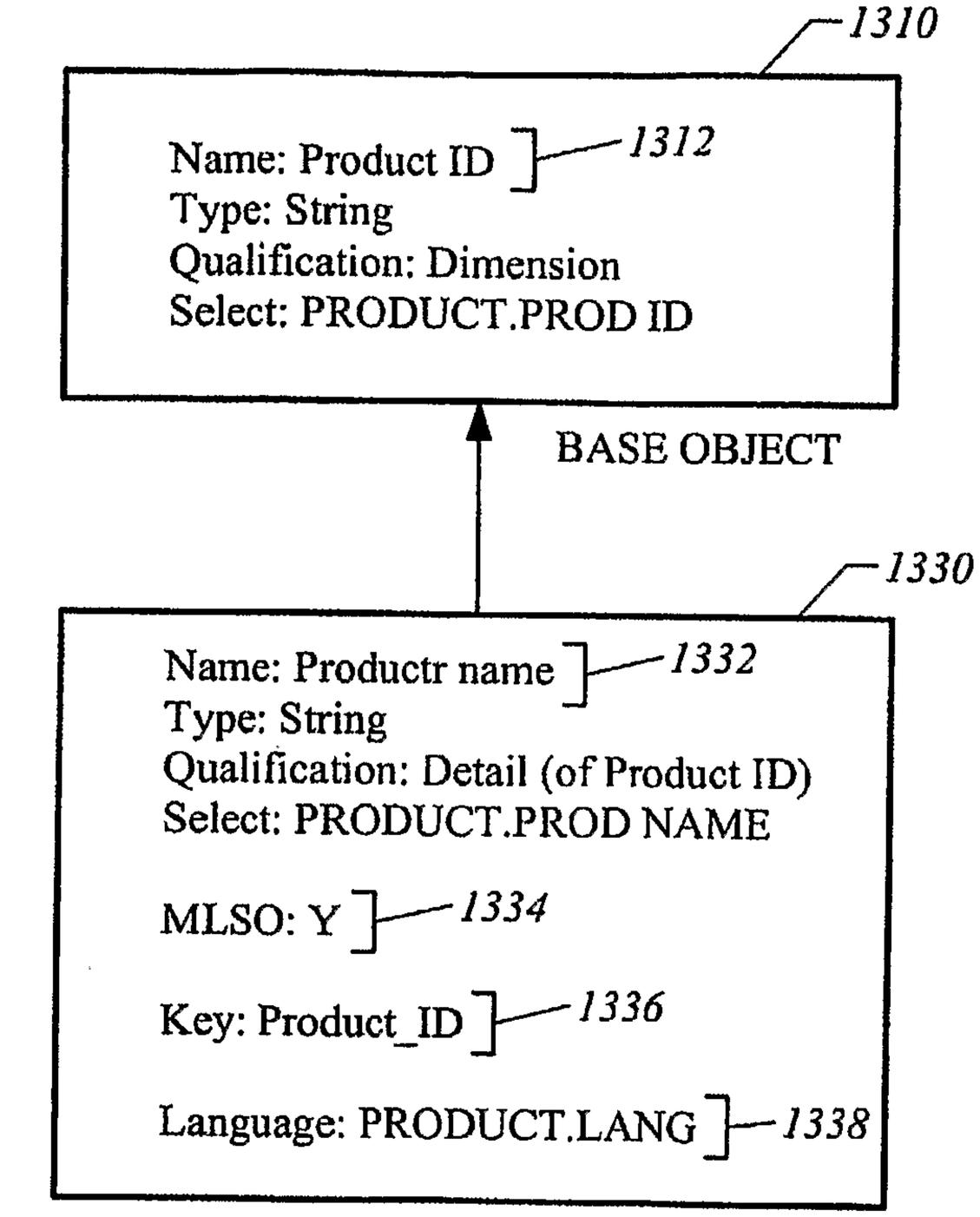
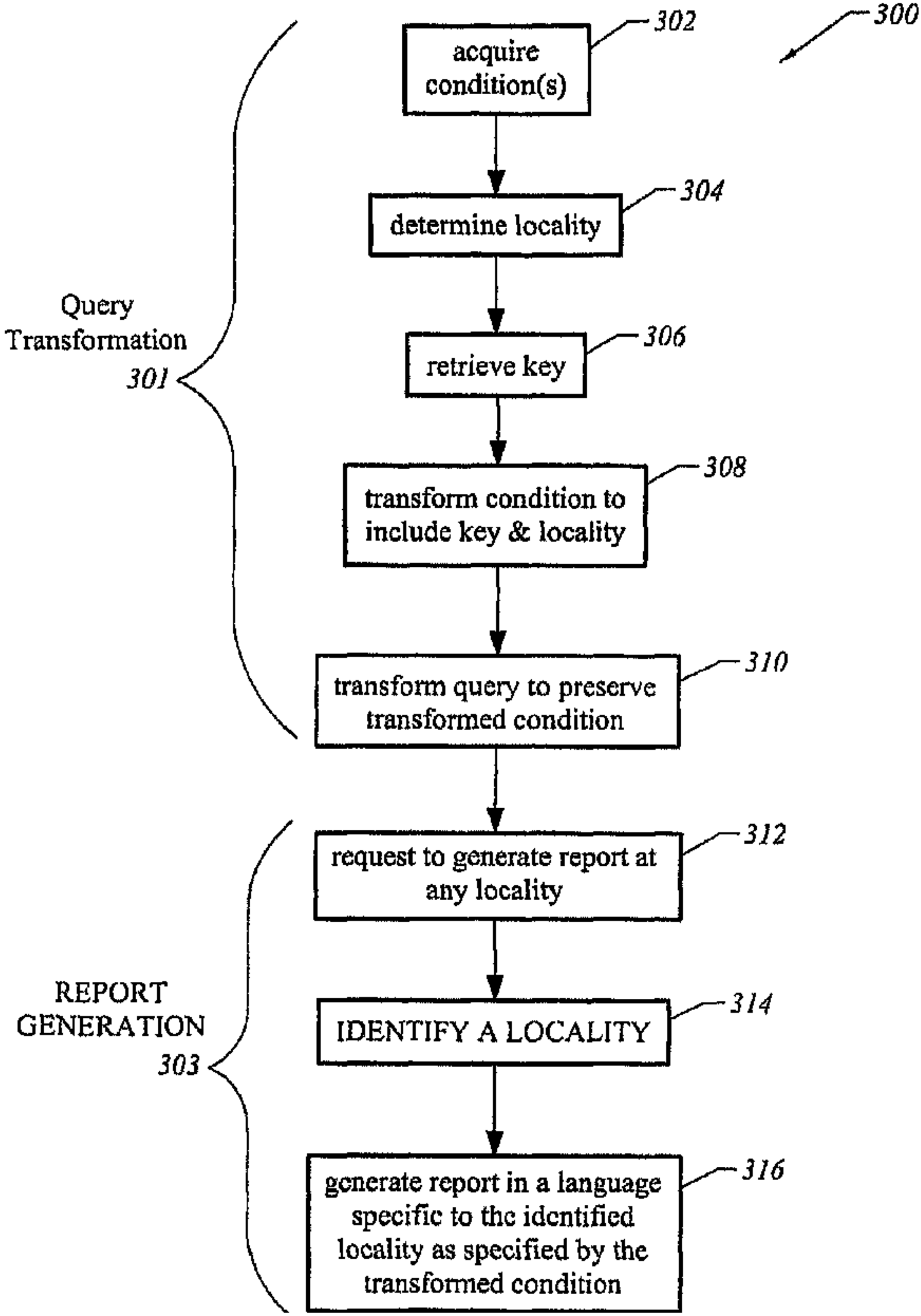

1310
Name: Product ID
1312
Type: String
Qualification: Dimension
Select: PRODUCT.PROD ID
BASE OBJECT
1330
Name: Productr name
1332
Type: String
Qualification: Detail (of Product ID)
Select: PRODUCT.PROD NAME
MLSO: Y
1334
Key: Product_ID
1336
Language: PRODUCT.LANG
1338
MULTILINGUAL SEMATIC LAYER OBJECT

FIG. 13


300
Query Transformation
301
REPORT GENERATION
303
acquire condition(s)
302
determine locality
304
retrieve key
306
transform condition to include key & locality
308
transform query to preserve transformed condition
310
request to generate report at any locality
312
IDENTIFY A LOCALITY
314
generate report in a language specific to the identified locality as specified by the transformed condition
316