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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 October 2017 (19.10.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/181131 A1

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

(21) International Application Number:
PCT/US2017/027791

(22) International Filing Date:
14 April 2017 (14.04.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/099,533 14 April 2016 (14.04.2016) US

(71) Applicant: **SALESFORCE.COM, INC.** [US/US]; The Landmark @ One Market, Suite 300, San Francisco, California 94105 (US).

(72) Inventors: **TIMMERMAN, Jan Michael**; 4364 Ontario Street, Vancouver, British Columbia V5V 3G9 (CA). **SCHNEIDER, Donovan**; 25 Aptos Avenue, San Francisco, California 94127 (US). **GITELMAN, Alex**; 1175 Oxford Street, Berkeley, California 94707 (US).

(74) Agents: **DURDIK, Paul** et al.; Haynes Beffel & Wofeld LLP, P.O. Box 366, 637 Main Street, Half Moon Bay, California 94019 (US).

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

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

Declarations under Rule 4.17:

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

[Continued on next page]

(54) Title: FINE GRAIN SECURITY FOR ANALYTIC DATA SETS

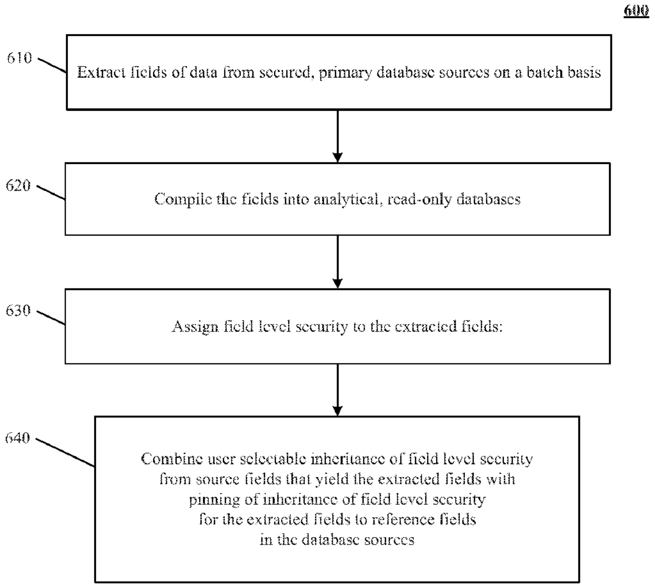

600
610
Extract fields of data from secured, primary database sources on a batch basis
620
Compile the fields into analytical, read-only databases
630
Assign field level security to the extracted fields:
640
Combine user selectable inheritance of field level security from source fields that yield the extracted fields with pinning of inheritance of field level security for the extracted fields to reference fields in the database sources

FIG. 6

(57) Abstract: The technology disclosed relates to assigning field level security to fields extracted from primary sources on a batch basis and compiled into analytical, read-only databases, for ultra-fast, ad-hoc data exploration and faceted navigation on integrated, heterogeneous data sets. The method includes assigning field level security to the extracted fields by combining user selectable inheritance of field level security from source fields that yield the extracted fields, with pinning of inheritance of field level security for the extracted fields to reference fields in the database sources wherein the reference fields are distinct from the extracted fields. The disclosed method also includes receiving additional fields as unsecured data sets, and assigning field level security to the additional fields, received by combining user selectable explicit specification of field level security for the received fields with pinning of inheritance of field level security for the received fields to reference fields in the database sources.

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

— *with amended claims (Art. 19(1))*

Published:

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

FINE GRAIN SECURITY FOR ANALYTIC DATA SETS

RELATED APPLICATION

[0001] This application claims priority to U.S. Patent Application No. 15/099,533 entitled "FINE GRAIN SECURITY FOR ANALYTIC DATA SETS" filed April 14, 2016 (Atty. Docket No. SALE 1147-1/1699US). The related application is hereby incorporated by reference for all purposes.

[0002] This application is related to U.S. Patent Application No. 14/512,230 entitled "ROW-LEVEL SECURITY INTEGRATION OF ANALYTICAL DATA STORE WITH CLOUD ARCHITECTURE" filed October 10th, 2014 (Atty. Docket No. SALE 1096-1/1451US). As described in that application, indexed fields in such read-only databases can be called dimensions, and quantity fields can be called measures. More details are found in the related application, which is hereby incorporated by reference for all purposes, as it also was incorporated in the priority application.

FIELD OF THE TECHNOLOGY DISCLOSED

[0003] The field of the disclosed technology is fine grain security for ultra-fast, ad-hoc data exploration and faceted navigation on integrated, heterogeneous data sets.

BACKGROUND

[0004] The subject matter discussed in the background section should not be assumed to be prior art merely as a result of its mention in the background section. Similarly, a problem mentioned in the background section or associated with the subject matter of the background section should not be assumed to have been previously recognized in the prior art. The subject matter in the background section merely represents different approaches, which in and of themselves may also correspond to implementations of the claimed inventions.

[0005] Businesses need the ability to query and to view query results in real time, for large data sets being analyzed, in order to make informed business decisions. An analytics dataset includes a materialized collection of fields that has been optimized for ad-hoc query analysis – to make possible a sub-second query response for any query. Security for the fields of the data set is of paramount importance for the businesses.

[0006] Business customers need data security for datasets from both internal content management systems (CRMs), which include many hundreds of custom data fields on some objects, and from external entities – including other enterprise systems. CRM data security is vital in many broad categories of enterprise businesses, including sales, service marketing, social networking communities, analytics, customized applications and the Internet of Things (IoT).

Data security is also paramount for data from external sources – which can be received via comma separated value (CSV) files directly uploaded to an analytics platform. Additionally, businesses require effective data security for datasets created by combining existing datasets.

[0007] Security requirements, for systems that provide business analytics "live" for large volumes of data, drive the need for the ability to control and restrict user access to data and functionality. Data security can include organization-level features such as when, where and how users can login; object-level permissions that determine what actions (e.g., create, read edit, delete) a user can perform on records of each object; and record-level permissions that grant access through default settings, sharing rules, etc. Additionally folder organization can be used to secure a variety of data for an enterprise, including reports, visualization designs, email templates and documents; folder access can be set to public, or access can be provided on a role-based basis. Data field-level permissions specify which fields a user can view and edit on records of an object (e.g., specifying visible and read-only) in one implementation.

[0008] When it is desirable to deploy analytics results via a single visualization lens or group of lenses displayed as a dashboard to users with different security profiles, then either data field level security (FLS) is needed; or else it becomes necessary to create copies of datasets with varying numbers of fields dependent on the permissions of each user or user profile. This requires a system of creating copies of lenses and modifying for the datasets, and then configuring application sharing to control user access to the datasets and lenses for which the user has security permissions. The second option of creating multiple similar datasets can quickly become a maintenance and security problem, and can cause a significant system performance impact. Alternatively, the first option of using field level security allows a single report to be shared across users with different security profiles, while ensuring that users see only the data fields to which they have been granted access.

[0009] In an example use case, access to a sensitive field like salary needs to be limited to the employee, a human resources (HR) administrator and executive management, and needs to disallow read access to all other users. Using FLS, a report author with an HR admin profile can create a report that includes employee salaries, and the report author can confidently share the report with all HR users with full assurance that non-HR admins will not see the field or its data.

[0010] In another use case, individuals' health data must be protected to comply with Health Insurance Portability and Accountability Act (HIPAA) privacy rules. In this example, critical-information viewers (such as doctors and physicians' assistants) of patient data have profiles that specify that they can view data fields that include diagnosis descriptions and prescription information – fields that an accountant in the business office of the medical facility would not have a need to see.

[0011] The disclosed technology relates to specifying and enforcing data field level security on integrated, heterogeneous data sets.

SUMMARY

[0012] A simplified summary is provided herein to help enable a basic or general understanding of various aspects of exemplary, non-limiting implementations that follow in the more detailed description and the accompanying drawings. This summary is not intended, however, as an extensive or exhaustive overview. Instead, the sole purpose of this summary is to present some concepts related to some exemplary non-limiting implementations in a simplified form as a prelude to the more detailed description of the various implementations that follow.

[0013] Disclosed systems and methods are usable to assign field level security to data fields extracted from primary sources on a batch basis and compiled into analytical, read-only databases. The method includes assigning field level security to the extracted fields by combining user selectable inheritance of field level security from source fields that yield the extracted fields, with pinning of inheritance of field level security for the extracted fields to reference fields in the database sources.

[0014] The disclosed method also includes receiving additional fields as unsecured data sets, and assigning field level security to the additional fields, received by combining user selectable explicit specification of field level security for the received fields with pinning of inheritance of field level security for the received fields to reference fields in the database sources.

[0015] Other aspects and advantages of the technology disclosed can be seen on review of the drawings, the detailed description and the claims, which follow.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The included drawings are for illustrative purposes and serve only to provide examples of possible structures and process operations for one or more implementations of this disclosure. These drawings in no way limit any changes in form and detail that may be made by one skilled in the art without departing from the spirit and scope of this disclosure. A more complete understanding of the subject matter may be derived by referring to the detailed description and claims when considered in conjunction with the following figures, wherein like reference numbers refer to similar elements throughout the figures.

[0017] **FIG. 1** illustrates an example business information and analytics architecture environment with field level security.

[0018] **FIG. 2** shows a block diagram for example data sources for an analytics cloud environment.

[0019] **FIG. 3** shows an example field level security attribute for a combination of two joined and transformed datasets.

[0020] **FIG. 4** shows an example block diagram with exchanged messages for a remote request.

[0021] **FIG. 5** shows and example dashboard with multiple lenses (prior art).

[0022] **FIG. 6** shows an overview of the flow for implementing security for data for fields of data from a secured database.

[0023] **FIG. 7** is a block diagram of an example multi-tenant computer system capable of implementing fine grain security for analytic data sets.

DETAILED DESCRIPTION

Introduction

[0024] The following detailed description is made with reference to the figures. Sample implementations are described to illustrate the technology disclosed, not to limit its scope, which is defined by the claims. Those of ordinary skill in the art will recognize a variety of equivalent variations on the description that follows.

[0025] Insight data analysis supports data exploration, dashboard building, and declarative representation of data visualizations. During exploration and replayed exploration, changes in data filtering, grouping and presentation format are animated, showing how a change redistributes data values. Singularly and in combination, these features can contribute to successful data analysis and presentation.

[0026] During single panel data exploration and replay, new data visualizations are animated as they are designed. Drilling down on a data segment, for instance, causes the original data segment to subdivide according to the selected regrouping and visually progress through animated subdivision growth and rearrangement into a more granular data visualization. This helps the analyst understand the data, and subsequently, explain important data segments to colleagues who are interested in the process as well as the numbers.

[0027] The disclosed methods for fine grain security for analytic data sets supports the use of a single dashboard that reports data to be shared across users with different security profiles, while ensuring that users see only the fields to which they have been granted access.

[0028] Examples of systems, apparatus, and methods according to the disclosed implementations are described in a "sales opportunity" context. The examples of sales contacts such as leads, prospects and accounts are used solely to add context and aid in the understanding of the disclosed implementations. In other instances, data with numerous elements may include medical system diagnoses and test results, insurance claims, customer service call routing, etc. or any data that would have a significant number of features. Other applications are possible, so the following examples should not be taken as definitive or limiting either in scope, context or setting. It will thus be apparent to one skilled in the art that implementations may be practiced in or outside the "sales opportunity" context.

Field Level Security Environment

[0029] **FIG. 1** illustrates one environment for implementing fine grain security for analytic data sets in a field level security environment **100,** which includes analytical read-only data store **102**, secured CRM data store **104**, external data store **142** and multi-tenant CRM computing services **106.** FLS filter **116,** in multi-tenant CRM computing services **106,** filters large data sets based on the field level security roles and permissions of a query requestor.

[0030] Analytical read-only data store **102** includes read-only datasets, with field level security attributes of multiple users, usable for querying and viewing query results in real time, for large data sets being analyzed. Secured CRM data store **104** includes datasets extracted from multi-tenant CRM computing services **106** on a batch basis, in one example. Field level security for the extracted fields can be assigned by combining user selectable inheritance of field level security from source fields that yield the extracted fields, with pinning of inheritance of field level security for the extracted fields to reference fields in the database source wherein the reference fields are distinct from the extracted fields. The data acquired (extracted) from large data repositories, along with associated field level security, can be compiled into analytical read-only data store **102**, and is usable to create "raw" datasets—read-only data structures for analytics--that can be augmented, transformed, flattened, etc. and published as customer-visible datasets for business entities.

[0031] External data store **142** can include data from sources that are not part of an enterprise's content management system. Examples of external systems include but are not limited to SAP™, ORACLE E-BUSINESS™, PEOPLESOFT™, NETSUITE™ and WORKDAY™. This data can include customer purchase history, demographics, relationships, and preferences. In one example, data can be represented as comma separated values (CSV) that represent sales quotas for a competitor, provided in a spreadsheet format. Field level security can be included in the spreadsheet data representation or can be user selectable by an administrator. External data store **142** can also include data received from external sources whose field level security is assigned to the additional received fields by combining user selectable explicit specification of field level security for the fields. Data can be received in other formats– including, but not limited to, other delimiter-separated formats, bitmapped images, Ogg format containers for different types of multimedia, and proprietary file formats.

[0032] The environment shown in **FIG. 1** includes analytics server **162** which handles heterogeneous data sources and related exploration, and analytics FLS engine **146** with internal FLS engine **156** for managing field level security metadata from multiple sources – including inherited, pinned and user-selected FLS. Additionally network **145** communicates among the data stores, servers, and engines described herein.

[0033] Also shown in the environment is GUI client engine **166** which includes visualization display engine **176** with lens and dashboard builder **178** for viewing live business analytics lenses and dashboards, and accepting new and updated query requests for exploring nuances of "live" analytic data. These requests can be routed to the analytics server **162** for service. Lens and dashboard builder **178** designs dashboards, displaying multiple lenses as real-time data query results. That is, an analyst can arrange display charts for multiple sets of query results on a single dashboard. When a change to a global filter affects any display chart on the dashboard, the remaining display charts on the dashboard get updated to reflect the change. Accurate live query results are produced and displayed across the set of display lenses on the dashboard, subject to the field level security permissions of the user requesting the query results. GUI client engine **166**, manages the flow of analytic query requests from, and responses to the analytics server **162**– for a user of mobile application **165 or** user computing device **164.**

[0034] A reference table with fields that have desired security profile attributes can be created, and fields in the analytic table can be pinned to fields in the reference table. That is, the field level security for fields can be inherited from a reference field in the table, with binding to existing columns and attributes in the data store.

[0035] GUI client engine **166** also includes a user-selectable FLS interface **186** usable by an administrator to select and configure security metadata options for fields in analytical read-only data store **102**.

[0036] Multi-tenant CRM computing services **106** can be of varying types including a workstation, server, computing cluster, blade server, server farm, or any other data processing system or computing device; and network **145** can be any network or combination of networks of devices that communicate with one another. For example, multi-tenant CRM computing services **106** can be implemented using one or any combination of a LAN (local area network), WAN (wide area network), telephone network (Public Switched Telephone Network (PSTN), Session

Initiation Protocol (SIP), 3G, 4G LTE), wireless network, point-to-point network, star network, token ring network, hub network, WiMAX, Wi-Fi, peer-to-peer connections like Bluetooth, Near Field Communication (NFC), Z-Wave, ZigBee, or other appropriate configuration of data networks, including the Internet. In other implementations, other networks can be used such as an intranet, an extranet, a virtual private network (VPN), a non-TCP/IP based network, any LAN or WAN or the like.

[0037] Analytical read-only data store **102**, secured CRM data store **104**, external data store **142**, and multi-tenant CRM computing services **106** can be implemented using a general-purpose distributed memory caching system. In some implementations, data structures can store information from one or more tenants into tables of a common database image to form an on-demand database service (ODDS), which can be implemented in many ways, such as a multi-tenant database system (MTDS). A database image can include one or more database objects. In other implementations, the databases can be relational database management systems (RDBMSs), object oriented database management systems (OODBMSs), distributed file systems (DFS), no-schema database, or any other data storing systems or computing devices. Analytical, read-only databases can implement response times of under two seconds when searching over twenty million records and compiling aggregate statistics from selected records.

[0038] In some implementations, user computing device **164** can be a personal computer, a laptop computer, tablet computer, smartphone or other mobile computing device, personal digital assistant (PDA), digital image capture devices, and the like. In some implementations, user mobile device **165** can be a tablet computer, smartphone or other mobile computing device, personal digital assistant (PDA), digital image capture devices, and the like.

[0039] Visualization display engine **176** and user-selectable FLS interface **186** can take one of a number of forms, running in a browser or as an application, including user interfaces, dashboard interfaces, engagement consoles, and other interfaces, such as mobile interfaces, tablet interfaces, summary interfaces, or wearable interfaces. In some implementations, it can be hosted on a web-based or cloud-based server in an on-premise environment. In one implementation, visualization display engine **176** and user-selectable FLS interface **186** can be accessed from a browser running on a computing device. The browser can be CHROME™, INTERNET EXPLORER™, FIREFOX™, SAFARI™, OPERA™, and the like. In other implementations, visualization display engine **176** and user-selectable FLS interface **186** can run as an engagement console on a computer desktop application.

[0040] In other implementations, field level security environment **100** may not have the same elements or components as those listed above and/or may have other/different elements or components instead of, or in addition to, those listed above, such as a web server and template database. The different elements or components can be combined into single software modules and multiple software modules can run on the same hardware. Data security is considered at multiple levels in the data flow for a system; one example is described next.

[0041] **FIG. 2** shows an example data flow diagram which includes, but is not limited to, CRM data **242** and external data **292**. For analytics cloud **245**, app **254** includes datasets **264** that incorporate data from CRM data **242** and external data **292.** Datasets **264** can be rendered as lenses **265**. Dashboards **266** can display multiple lenses **265** for delivery of analytics based on CRM data **242** and external data **292** to users **268**.

[0042] In one use case, an administrator for the enterprise controls access to the company's CRM data **242**. Dataset owners control access to records in datasets **264**. App owners, administrators, and users with sufficient permissions (e.g., granted manager access to an app) control access to datasets **264**, lenses **265** and dashboards **266** within the apps. The disclosed methods of delivering fine grain security via field level security can ensure that users **268** see only the data for which they have permissions. Security can be adapted to an extracted dataset that is detached at build time from the CRM relational database.

[0043] Security can also be adapted to handle data extracted from external sources, operating under different infrastructures and security models, and combined with the extracted CRM data. In one example use case, a large enterprise may be exploring boat sales by month for competing boat brands, while analyzing recent loan data extracted from a data integration enterprise whose research data targets chief intelligence officers (CIOs) senior IT leads, marketing leaders and supply chain leaders. That is, heterogeneous data from multiple external sources can be included on a dashboard for users whose profile covers the needed field level security.

[0044] For some use cases, multiple datasets need to be joined and transformed, and the field level security for resultant fields can be calculated based on a combination of field level security settings in the two or more objects whose data has been joined.

[0045] The following discussion motivates the use of joins and transformations of datasets for facilitating the delivery of "live" data analytics that can illuminate business quandaries. Various types of on-demand transactional data management systems can be integrated with analytic data stores to provide to data analysts ad hoc access to query the transaction data management systems. This can facilitate rapid building of analytic applications that use numerical values, metrics and measurements to drive business intelligence from transactional data stored in the transaction data management systems and support organizational decision making. Transaction data refers to data objects that support operations of an organization and are included in application systems that automate key business processes in different areas such as sales, service, banking, order management, manufacturing, medical records, purchasing, billing, etc. Some examples of transaction data include enterprise data (e.g. order-entry, supply-chain, shipping, invoices), sales data (e.g. accounts, leads, opportunities), medical (diagnoses, prescriptions, billing), and the like.

[0046] Extraction refers to the task of acquiring transaction data from transactional data stores, according to one implementation. This can be as simple as downloading a flat file from a database or a spreadsheet, or as sophisticated as setting up relationships with external systems that then control the transportation of data to the target system. Loading is the phase in which the captured data is deposited into a new data store such as a warehouse or a mart. In some implementations, loading can be accomplished by custom programming commands such as IMPORT in structured query language (SQL) and LOAD in Oracle Utilities. In some implementations, a plurality of application-programming interfaces (APIs) can be used, to interface with a plurality of transactional data sources, along with extraction connectors that load the transaction data into dedicated data stores.

[0047] Transformation refers to the stage of applying a series of rules or functions to the extracted or the loaded data, generally so as to convert the extracted or the loaded data to a format that is conducive for deriving analytics. Some examples of transformation include selecting only certain columns to load, translating coded values, encoding free-form values, deriving new calculated values, sorting, joining data from multiple sources, aggregation, de-normalization, transposing or pivoting data, splitting a column into multiple columns and data validation.

[0048] In one implementation, an augment transformation joins data from two datasets to enable queries across both of them. For instance, augmenting a “user dataset” with an “account dataset” can enable a data analyst to generate query that displays all account details, including the names of the account owner and creator. Augmentation transformation creates a new dataset based on data from two input dataset. Each input dataset can be identified as the left or right dataset. The new dataset includes all the columns of the left dataset and appends only the specified columns from the right dataset. Augmentation transformation performs a left, outer join, where the new dataset includes all rows from the left dataset and only matched rows from the right dataset. In another implementation, queries can be enabled that span more than two dataset. This can be achieved by augmenting two datasets at a time. For example, to augment three datasets, a first two datasets can be augmented before augmenting the resulting dataset with a third dataset.

[0049] In some implementations, a join condition in the augment transformation can be specified to determine how to match rows in the right dataset to those in the left dataset. The following example illustrates a single-column join condition. To augment the following datasets based on single-column key, an "Opportunity" is assigned as the left dataset and an "Account" is assigned as the right dataset. Also, "OpptyAcct" is specified as the relationship between them.

Opportunity dataset	Account dataset
ID	*ID
Opportunity_Name	Account_Name
Amount	Annual_Revenue
Stage	Billing_Address
Closed_Date	
*Account_ID	

[0050] An "OpptyAcct" prefix is added to all account columns and the datasets are joined based on a key defined as "Opportunity.Account_ID=Account.ID." After augmenting the two input datasets, the resulting dataset includes the following columns:

Opportunity-Account EdgeMart

ID
Opportunity_Name
Amount
Stage
Closed_Date
Account_ID

OpptyAcct.Account_Name
OpptyAcct.Annual_Revenue
OpptyAcct.Billing_Address

[0051] In other implementations, different heavy-weight transformations can be applied, including a flatten transformation to create role-based access on accounts, index transformation to index one dimension column in an dataset, Ngram transformation to generate case-sensitive, full-text index based on data in an dataset, register transformation to register an dataset to make it available for queries and extract transformation to extract data from fields of a data object.

[0052] For some implementations, fine grain security for analytical data sets includes. joining data from two or more objects in the primary database sources; and calculating field level security in one or more of the fields in the joined data based on a combination of field level security settings in the two or more objects.

[0053] **FIG. 3** shows an example implementation of joining two datasets **300.** Augment engine **345** can join two objects: dataset A **324** and dataset B **328**. In some implementations, a new field can be calculated from data in two or more fields in the primary database sources; and calculating field level security in the new field based on a combination of field level security settings in the two or more objects.

[0054] Transform engine **355** can change the results of the join. In one implementation the field level security for the resulting fields, after the join and transform, are based on the field level security settings in dataset A **324** and dataset B **328**. Resultant dataset **365** includes field X with security level set to by aggregating the security **344**, **346** of the fields that make up the derived field. In one example, field A has FLS dependent on column one, field B has FLS dependent on column two, and field C is derived from Field A and Field B –depending on both and bound to both, with field visibility dependent on both column one and column two. Continuing with the example, the administrator can be allowed to override this default setting: manually binding field C FLS to column one, or to column two, or to a totally different column (column three). Displaying the list of columns (one, two or 'other') would be based on the lineage information we have for field C.

[0055] Override options are useful when aggregate statistics are more widely distributed than underlying details. Roll-ups may not inherit the security restriction of the field that they summarize. Special derived security for fields can be produced from JOIN operations. Special derived security for calculated fields may deserve less security than one of the components.

[0056] An example block diagram in **FIG. 4** displays one implementation with customized field level security requirements for users– showing requests and responses between functional blocks in the environment. GUI client engine **166** sends a request (e.g. a list of lenses, a list of datasets, a remote query for external data). Upon receipt of the request, analytics server **162** determines whether a new query is needed, and if needed, sends remote request **424** to multi-tenant CRM computing services **106,** which sends a request to the internal FLS engine **156** in analytics FLS engine **146** – a request for the FLS requirements **434** for the requesting user. When the composite FLS **444** for the user is returned to FLS filter **116**, then the FLS-filtered response **454** gets generated and sent to analytics server **162** which serves the query results for display to the user via GUI client engine **166**.

[0057] For some implementations, FLS can be specified for fields in a query. In one use case, an analytics server request includes the objects of interest, along with the field lineage for those objects. The analytics server can respond with an object which includes the fields that are allowed for the requesting user, based on their authorized FLS. The queries and responses can be delivered using JSON or another format that transmits data objects consisting of attribute-value pairs.One example of field lineage can be expressed as shown below.

```
" field3.relation1.relation2.field "
    "<entity1>" : [
        "<field1>",
        "<field2>",
        "<field3.relation1.relation2.field>",
        ...]
    "<entity2>" : [
        "<field1>",
        "<field2>",
        "<field3.relation1.relation2.field>",
        ...]
```

[0058] A user selectable FLS interface **186** can provide a screen interface for an administrator to explicitly specify field level security for extracted fields. The specified FLS, by the administrator, can override inheritance of field level security for the extracted fields.

[0059] When a user requests a lens, or a dashboard of multiple lenses, that calls for a field for which the user does not have the required field level security profile, the GUI can deliver a message that conveys to the user that their request cannot be completed. For some implementations, if any field needed for rendering the lens is not viewable by the user, then the lens is not delivered, and a UI that communicates “lens not visible” can be displayed. This result can occur in cases when the user is requesting display of a list of available lenses, and also in cases when the user requests to run a lens query. Similarly, if a user has one or more lenses for which they lack security permissions, then the dashboard can hide the lens, in some implementations, and can display a message that communicates that the lens is hidden from the user in other cases.

[0060] In some use cases, when a user lacks the required FLS profile, the fields for which they lack security permissions can be pruned from the available fields for the project, and a subset of the fields can be used to create a runnable query. Visualization success may be limited in this scenario.

[0061] In another implementation, a list of fields needed to complete a user's query can be included in the FLS-filtered response to analytics server **162** and queries for lenses and dashboards can be filtered based on the returned list of allowable fields. The analytics server **162** can inspect the fields needed to satisfy the queries in the stored visualizations: lenses and dashboards. The list of lenses and dashboards provided to the GUI **166** can then be filtered by the analytics server **162**. This will prevent users from attempting to view lenses and dashboards which will error out and cause display of a "lens not visible" message. For situations in which an advanced user attempts to type in a field name for which they do not have field level security access, the system does not acknowledge recognition of the field name. The disclosed technology includes a specification for extracting the fields of data that includes an advanced search comprising a user-entered string recognized as a field to which the user does not have field level access and treating the recognized string as not recognized.

[0062] **FIG. 5** (prior art) is shown as an example of a dashboard with multiple lenses. In this visualization, a user viewing the dashboard is not allowed to access the "Account Name" field. The dashboard includes two widgets which depend on this field: the right most filter selector - "account selector" **525** and the bottom right pie chart **568**. Two display options include filtering these two components out of the dashboard, or rendering the dashboard and displaying an error message in place of those two lenses. As a third alternative, the user could be prevented from being able to view the dashboard at all.

[0063] In some implementations, a dataset can be created and cached, and the user's session can use the FLS information that existed when the dataset for the session was created. For this use case, the field level security restrictions that a user has when they begin a session will continue throughout the session.

[0064] A number of options are available to respond to changes in FLS that occur between extract/build cycles for security analytics. When cache time is zero, that is no caching at all, full security fidelity is achieved, with no security drift. In some implementations, caching of current values of the field level security for the fields in the analytical read-only database is valid for a limited time of one of 5-10 minutes, before querying for an update to the current values.

System Flow

[0065] **FIG. 6** illustrates a flowchart of one implementation **600** of implementing fine grain security for analytic data sets. Flowchart **600** can be implemented at least partially with a database system, e.g., by one or more processors configured to receive or retrieve information, process the information, store results, and transmit the results. Other implementations may perform the steps in different orders and/or with different, fewer or additional steps than the ones illustrated in **FIG. 6**. The actions described below can be subdivided into more steps or combined into fewer steps to carry out the method described using a different number or arrangement of steps.

[0066] At action **610**, the FLS engine **116** extracts fields of data from secured, primary database sources on a batch basis.

[0067] At action **620,** the analytics FLS engine **146** compiles the fields into analytical read only data store **102**.

[0068] At action **630**, internal FLS engine **156** assigns field level security to the extracted fields, and can manage field level security metadata from multiple sources – including inherited, pinned and user-selected FLS.

[0069] At action **640** analytics FLS engine **146** combines user selectable inheritance of field level security from source fields that yield the extracted fields with pinning of inheritance of field level security for the extracted fields to reference fields in the database sources.

Multi-Tenant Integration

[0070] **FIG. 7** presents a block diagram of an exemplary multi-tenant system **700** suitable for integration with field level security environment **100** of **FIG. 1**. In general, the illustrated multi-tenant system **700** of **FIG. 7** includes a server **704** that dynamically creates and supports virtual applications **716** and **718,** based upon data **722** from a common database **732** that is shared between multiple tenants, alternatively referred to herein as a "multi-tenant database". Data and services generated by the virtual applications **716** and **718,** including GUI clients, are provided via a network **745** to any number of client devices **748** or **758**, as desired.

[0071] As used herein, a "tenant" or an "organization" refers to a group of one or more users that shares access to common subset of the data within the multi-tenant database **732**. In this regard, each tenant includes one or more users associated with, assigned to, or otherwise belonging to that respective tenant. Stated another way, each respective user within the multi-tenant system **700** is associated with, assigned to, or otherwise belongs to a particular tenant of the plurality of tenants supported by the multi-tenant system **700**. Tenants may represent users, user departments, work or legal organizations, and/or any other entities that maintain data for particular sets of users within the multi-tenant system **700**. Although multiple tenants may share access to the server **704** and the database **732**, the particular data and services provided from the server **704** to each tenant can be securely isolated from those provided to other tenants. The multi-tenant architecture therefore allows different sets of users to share functionality and hardware resources without necessarily sharing any of the data **722** belonging to or otherwise associated with other tenants.

[0072] The multi-tenant database **732** is any sort of repository or other data storage system capable of storing and managing the data **722** associated with any number of tenants. The database **732** may be implemented using any type of conventional database server hardware. In various implementations, the database **732** shares processing hardware with the server **704**. In other implementations, the database **732** is implemented using separate physical and/or virtual database server hardware that communicates with the server **704** to perform the various functions described herein. The multi-tenant database **732** may alternatively be referred to herein as an on-demand database, in that the multi-tenant database **732** provides (or is available to provide) data at run-time to on-demand virtual applications **716** or **718** generated by the application platform **717**, with tenant1 metadata **712** and tenant2 metadata **714** securely isolated.

[0073] In practice, the data **722** may be organized and formatted in any manner to support the application platform **717**. In various implementations, conventional data relationships are established using any number of pivot tables **713** that establish indexing, uniqueness, relationships between entities, and/or other aspects of conventional database organization as desired.

[0074] The server **704** is implemented using one or more actual and/or virtual computing systems that collectively provide the dynamic application platform **717** for generating the virtual applications. For example, the server **704** may be implemented using a cluster of actual and/or virtual servers operating in conjunction with each other, typically in association with conventional network communications, cluster management, load balancing and other features as appropriate. The server **704** operates with any sort of conventional processing hardware such as a processor **736**, memory **738**, input/output features **734** and the like. The input/output **734** generally represent the interface(s) to networks (e.g., to the network **745**, or any other local area, wide area or other network), mass storage, display devices, data entry devices and/or the like. User interface input devices **734** can include a keyboard; pointing devices such as a mouse, trackball, touchpad, or graphics tablet; a scanner; a touch screen incorporated into the display; audio input devices such as voice recognition systems and microphones; and other types of input devices. In general, use of the term "input device" is intended to include possible types of devices and ways to input information into computer system **717**.

[0075] User interface output devices can include a display subsystem, a printer, a fax machine, or non-visual displays such as audio output devices. The display subsystem can include a cathode ray tube (CRT), a flat-panel device such as a liquid crystal display (LCD), a projection device, or some other mechanism for creating a visible image. The display subsystem can also provide a non-visual display such as audio output devices. In general, use of the term "output device" is intended to include all possible types of devices and ways to output information from processor **736** to the user or to another machine or computer system.

[0076] The processor **736** may be implemented using any suitable processing system, such as one or more processors, controllers, microprocessors, microcontrollers, processing cores and/or other computing resources spread across any number of distributed or integrated systems, including any number of "cloud-based" or other virtual systems. The memory **738** represents any non-transitory short or long term storage or other computer-readable media capable of storing programming instructions for execution on the processor **736**, including any sort of random access memory (RAM), read only memory (ROM), flash memory, magnetic or optical mass storage, and/or the like. The computer-executable programming instructions, when read and executed by the server **704** and/or processor **736**, cause the server **704** and/or processor **736** to create, generate, or otherwise facilitate the application platform **717** and/or virtual applications **716** and **718,** and perform one or more additional tasks, operations, functions, and/or processes described herein. It should be noted that the memory **738** represents one suitable implementation of such computer-readable media, and alternatively or additionally, the server **704** could receive and cooperate with external computer-readable media that is realized as a portable or mobile component or application platform, e.g., a portable hard drive, a USB flash drive, an optical disc, or the like.

[0077] The application platform **717** is any sort of software application or other data processing engine that generates the virtual applications **716** and **718** that provide data and/or services to the client devices **748** and **758**. In a typical implementation, the application platform **717** gains access to processing resources, communications interfaces and other features of the processing hardware using any sort of conventional or proprietary operating system **728**. The virtual applications **716** and **718** are typically generated at run-time in response to input received from the client devices **748** and **758**.

[0078] With continued reference to **FIG. 7**, the data and services provided by the server **704** can be retrieved using any sort of personal computer, mobile telephone, tablet or other network-enabled client device **748** or **758** on the network **745**. In an exemplary implementation, the client device **748** or **758** includes a display device, such as a monitor, screen, or another conventional electronic display capable of graphically presenting data and/or information retrieved from the multi-tenant database **732**.

[0079] In some implementations, network(s) **745** can be any one or any combination of Local Area Network (LAN), Wide Area Network (WAN), WiMAX, Wi-Fi, telephone network, wireless network, point-to-point network, star network, token ring network, hub network, mesh network, peer-to-peer connections like Bluetooth, Near Field Communication (NFC), Z-Wave, ZigBee, or other appropriate configuration of data networks, including the Internet.

[0080] The foregoing description is merely illustrative in nature and is not intended to limit the implementations of the subject matter or the application and uses of such implementations. Furthermore, there is no intention to be bound by any expressed or implied theory presented in the technical field, background, or the detailed description. As used herein, the word "exemplary" means "serving as an example, instance, or illustration." Any implementation described herein as exemplary is not necessarily to be construed as preferred or advantageous over other implementations, and the exemplary implementations described herein are not intended to limit the scope or applicability of the subject matter in any way.

[0081] The technology disclosed can be implemented in the context of any computer-implemented system including a database system, a multi-tenant environment, or a relational database implementation like an ORACLE™ compatible database implementation, an IBM DB2 Enterprise Server compatible relational database implementation, a MySQL or PostgreSQL compatible relational database implementation or a Microsoft SQL Server compatible relational database implementation or a NoSQL non-relational database implementation such as a Vampire™ compatible non-relational database implementation, an Apache Cassandra™ compatible non-relational database implementation, a BigTable compatible non-relational database implementation or an HBase or DynamoDB compatible non-relational database implementation.

[0082] Moreover, the technology disclosed can be implemented using two or more separate and distinct computer-implemented systems that cooperate and communicate with one another. The technology disclosed can be implemented in numerous ways, including as a process, a method, an apparatus, a system, a device, a computer readable medium such as a computer readable storage medium that stores computer readable instructions or computer program code, or as a computer program product comprising a computer usable medium having a computer readable program code embodied therein.

Particular Implementations

[0083] In one implementation, a disclosed method includes extracting fields of data from secured, primary database sources on a batch basis and compiling the fields into analytical, read-only databases; and assigning field level security to the extracted fields by combining user selectable inheritance of field level security from source fields that yield the extracted fields, with pinning of inheritance of field level security for the extracted fields to reference fields in the database sources wherein the reference fields are distinct from the extracted fields. The disclosed method can further include receiving additional fields as unsecured data sets; and assigning field level security to the additional fields received by combining user selectable explicit specification of field level security for the received fields, with pinning of inheritance of field level security for the received fields to reference fields in the database sources wherein the reference fields are distinct from the extracted fields.

[0084] This method and other implementations of the technology disclosed can include one or more of the following features and/or features described in connection with additional methods disclosed. In the interest of conciseness, the combinations of features disclosed in this application are not individually enumerated and are not repeated with each base set of features. The reader will understand how features identified in this section can readily be combined with sets of base features identified as implementations.

[0085] For some implementations, the user selectable assigning of field level security further includes assigning field level security by overriding inheritance of field level security from the extracted fields with explicit specification of field level security.

[0086] Implementations of the disclosed method for analytical, read-only databases implement response times of under two seconds when searching over twenty million records and compiling aggregate statistics from selected records.

[0087] For some implementations, the disclosed method of fine grain security for analytical data sets includes joining data from two or more objects in the primary database sources; and calculating field level security in one or more of the fields in the joined data based on a combination of field level security settings in the two or more objects.

[0088] In yet other implementations the method can include flagging user-selected reference fields in the database sources as a basis for the pinning; and causing display of a user interface that lists the flagged reference fields and enables user pinning of the extracted fields to the reference fields.

[0089] Implementations of the disclosed method further include a specification for extracting the fields of data that includes an advanced search comprising a user-entered string recognized as a field to which the user does not have field level access and treating the recognized string as not recognized.

[0090] For some implementations of the disclosed method, when a requested lens uses a field for which a user does not have field level security permission, the lens is not displayed; in some cases the display includes a message that the lens is not available. Handling of a dashboard that calls for a field that isn't available to a user can include a subset of lenses for the dashboard for which the user has field level security permission, and a message that communicates that the remaining lens is not available for viewing. In an implementation that includes an advance query specified by a string that calls for a field that isn't available to a user, no protected data for which the requesting user does not have security permission is displayed.

[0091] For yet other implementations, caching of user permissions for fields in the analytical read-only database is for a limited time including but not limited to one of 2-6 minutes, 3-8 minutes, or 4-10 minutes before refreshing the user permissions.

[0092] The disclosed method can further include caching current values of the field level security for the fields in the analytical read-only database for a limited time of one of 3-8 minutes, before querying for an update to the current values. In other implementations the method can include caching current values of the field level security for the fields in the analytical read-only database for a limited time of one of 4-9 minutes, before querying for an update to the current values; or can include caching current values of the field level security for the fields in the analytical read-only database for a limited time of one of 5-10 minutes, before querying for an update to the current values.

[0093] Other implementations may include non-transitory tangible computer readable media impressed with instructions that, when executed on a computer device and one or more servers, perform any of the processes described above.

[0094] Yet another implementation may include a computing system including at least one server comprising one or more processors and memory, coupled to the processors, containing computer instructions that, when executed on the processors, cause the computing system to perform any of the processes described above.

[0095] While the technology disclosed is disclosed by reference to the preferred embodiments and examples detailed above, it is to be understood that these examples are intended in an illustrative rather than in a limiting sense. It is contemplated that modifications and combinations will readily occur to those skilled in the art, which modifications and combinations will be within the spirit of the invention and the scope of the following claims.

Clauses

1. We disclose a system for building a secure read-only database, including:

extraction means for batch extraction of data fields comprising dimensions and measures from at least one primary database source;

compiling means for compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

security assignment means for assigning data field level security access to a plurality of the compiled data fields, the security means comprising selection means for a selection between security inheritance and security pinning, wherein

selection of the security inheritance causes the security assignment means to assign data field level security by inheritance, from an extracted field in the primary database sources, to the compiled data field; and

selection of the security pinning causes the security assignment means to assign data field level security by security pinning, from a reference field to the compiled data field, with inheritance of the data field level security from the pinned reference field to the compiled data field;

wherein reference field is distinct from the extracted field;

building means for correlating the assigned data field level security with the compiled data fields during building of the read-only database.

2. The system of clause 1, further including:

the extraction means further for extraction of additional data fields from unsecured data sets that are not subject to data filed level security; and

security assignment means further for assigning field level security to the additional data fields with section means for further selecting between explicit security and security pinning, wherein

selection of the explicit security causes the security assignment means to assign data field level security by explicit specification of field level security for the additional data fields, and

selection of the security pinning causes the security assignment means to assign data field level security by security pinning, from a reference field in a reference table to the compiled data field, with inheritance of the data field level security from the pinned reference field to the additional data fields.

3. The system of clause 2, the selection means further for selecting between explicit security and selection of explicit security causes the security assignment means to assign data field level security by explicit specification of field level security for the additional data fields.

4. The system of any of clauses 1-3, wherein the read-only databases implement response times of under two seconds when searching over twenty million records and compiling aggregate statistics from selected records.

5. The system of any of clauses 1-4, further including:

a new field means for calculating a new data field from data in two or more data fields in the primary database source; and

the security assignment means further for calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

6. The system of any of clauses 1-5, further including:

a object joining means for joining data from two or more objects in the primary database source; and

the security assignment means further for calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

7. The system of any of clauses 1-6, further including:

the selection means further for flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

8. We further disclose a method of securing a read-only database, including:

extracting data fields comprising dimensions and measures from at least one primary database source;

compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

assigning data field level security access to a plurality of the compiled data fields with selection between security inheritance and security pinning, wherein selection of the security inheritance causes assigning of data field level security by inheritance, from an extracted field in the primary database sources, to the compiled data field; and

selection of the security pinning causes assigning of data field level security by security pinning, from a reference field to the compiled data field, with inheritance of the data field level security from the pinned reference field to the compiled data field;

wherein reference field is distinct from the extracted field; and

correlating the assigned data field level security with the compiled data fields during building of the read-only database.

9. The method of clause 8, further including:

extracting additional data fields from at least one unsecured data set that is not subject to data filed level security; and

assigning field level security to the additional data fields for further selecting between explicit security and security pinning, wherein

selection of the explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields, and

selection of the security pinning causes assigning data field level security by security pinning, from a reference field in a reference table to the compiled data field, with inheritance of the data field level security from the pinned reference field to the additional data fields.

10. The method of clause 9, further including selecting between explicit security and selection of explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields.

11. The method of any of clauses 8-10, further including:

calculating a new data field from data in two or more data fields in the primary database source; and

calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

12. The method of any of clauses 8-11, further including:

joining data from two or more objects in the primary database source; and

calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

13. The method of any of clauses 8-12, further including:

flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

14. A non-transitory computer readable storage medium impressed with computer program instructions to secure a read-only database, the instructions, when executed on a processor, implement a method comprising:

extracting data fields comprising dimensions and measures from at least one primary database source;

compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

assigning data field level security access to a plurality of the compiled data fields with selection between security inheritance and security pinning, wherein

selection of the security inheritance causes assigning of data field level security by inheritance, from an extracted field in the primary database sources, to the compiled data field; and

selection of the security pinning causes assigning of data field level security by security pinning, from a reference field to the compiled data field, with inheritance of the data field level security from the pinned reference field to the compiled data field;

wherein reference field is distinct from the extracted field; and

correlating the assigned data field level security with the compiled data fields during building of the read-only database.

15. The non-transitory computer readable storage medium of clause 14, implementing the method further including:

extracting additional data fields from at least one unsecured data set that is not subject to data filed level security; and

assigning field level security to the additional data fields for further selecting between explicit security and security pinning, wherein

selection of the explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields, and

selection of the security pinning causes assigning data field level security by security pinning, from a reference field in a reference table to the compiled data field, with inheritance of the data field level security from the pinned reference field to the additional data fields.

16. The non-transitory computer readable storage medium of clause 15, implementing the method further including selecting between explicit security and selection of explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields.

17. The non-transitory computer readable storage medium of any of clauses 14-16, implementing the method further including:

calculating a new data field from data in two or more data fields in the primary database source; and

calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

18. The non-transitory computer readable storage medium of any of clauses 14-17, implementing the method further comprising:

joining data from two or more objects in the primary database source; and

calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

19. The non-transitory computer readable storage medium of any of clauses 14-18, implementing the method further comprising:

flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

CLAIMS

What is claimed is:

1. A system for building a secure read-only database, including:

extraction means for batch extraction of data fields comprising dimensions and measures from at least one primary database source;

compiling means for compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

security assignment means for assigning data field level security access to a plurality of the compiled data fields, the security means comprising selection means for a selection between security inheritance and security pinning, wherein

selection of the security inheritance causes the security assignment means to assign data field level security by inheritance, from an extracted field in the primary database sources, to the compiled data field; and

selection of the security pinning causes the security assignment means to assign data field level security by security pinning, from a reference field to the compiled data field, with inheritance of the data field level security from the pinned reference field to the compiled data field;

wherein reference field is distinct from the extracted field;

building means for correlating the assigned data field level security with the compiled data fields during building of the read-only database.

2. The system of claim 1, further including:

the extraction means further for extraction of additional data fields from unsecured data sets that are not subject to data filed level security; and

security assignment means further for assigning field level security to the additional data fields with section means for further selecting between explicit security and security pinning, wherein

selection of the explicit security causes the security assignment means to assign data field level security by explicit specification of field level security for the additional data fields, and

selection of the security pinning causes the security assignment means to assign data field level security by security pinning, from a reference field in a reference table to the compiled data field, with inheritance of the data field level security from the pinned reference field to the additional data fields.

3. The system of claim 2, the selection means further for selecting between explicit security and selection of explicit security causes the security assignment means to assign data field level security by explicit specification of field level security for the additional data fields.

4. The system of any of claims 1-3, wherein the read-only databases implement response times of under two seconds when searching over twenty million records and compiling aggregate statistics from selected records.

5. The system of any of claims 1-4, further including:

a new field means for calculating a new data field from data in two or more data fields in the primary database source; and

the security assignment means further for calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

6. The system of any of claims 1-5, further including:

a object joining means for joining data from two or more objects in the primary database source; and

the security assignment means further for calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

7. The system of any of claims 1-6, further including:

the selection means further for flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

8. A method of securing a read-only database, including:

extracting data fields comprising dimensions and measures from at least one primary database source;

compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

assigning data field level security access to a plurality of the compiled data fields with selection between security inheritance and security pinning, wherein

selection of the security inheritance causes assigning of data field level security by inheritance, from an extracted field in the primary database sources, to the compiled data field; and

selection of the security pinning causes assigning of data field level security by security pinning, from a reference field to the compiled data field, with inheritance of the data field level security from the pinned reference field to the compiled data field;

wherein reference field is distinct from the extracted field; and

correlating the assigned data field level security with the compiled data fields during building of the read-only database.

9. The method of claim 8, further including:

extracting additional data fields from at least one unsecured data set that is not subject to data filed level security; and

assigning field level security to the additional data fields for further selecting between explicit security and security pinning, wherein

selection of the explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields, and

selection of the security pinning causes assigning data field level security by security pinning, from a reference field in a reference table to the compiled data field, with inheritance of the data field level security from the pinned reference field to the additional data fields.

10. The method of claim 9, further including selecting between explicit security and selection of explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields.

11. The method of any of claims 8-10, further including:

calculating a new data field from data in two or more data fields in the primary database source; and calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

12. The method of any of claims 8-11, further including:

joining data from two or more objects in the primary database source; and

calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

13. The method of any of claims 8-12, further including:

flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

14. A non-transitory computer readable storage medium impressed with computer program instructions to secure a read-only database, the instructions, when executed on a processor, implement a method comprising:

extracting data fields comprising dimensions and measures from at least one primary database source;

compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

assigning data field level security access to a plurality of the compiled data fields with selection between security inheritance and security pinning, wherein

selection of the security inheritance causes assigning of data field level security by inheritance, from an extracted field in the primary database sources, to the compiled data field; and

selection of the security pinning causes assigning of data field level security by security pinning, from a reference field to the compiled data field, with inheritance of the data field level security from the pinned reference field to the compiled data field;

wherein reference field is distinct from the extracted field; and

correlating the assigned data field level security with the compiled data fields during building of the read-only database.

15. The non-transitory computer readable storage medium of claim 14, implementing the method further including:

extracting additional data fields from at least one unsecured data set that is not subject to data filed level security; and

assigning field level security to the additional data fields for further selecting between explicit security and security pinning, wherein

selection of the explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields, and

selection of the security pinning causes assigning data field level security by security pinning, from a reference field in a reference table to the compiled data field, with inheritance of the data field level security from the pinned reference field to the additional data fields.

16. The non-transitory computer readable storage medium of claim 15, implementing the method further including selecting between explicit security and selection of explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields.

17. The non-transitory computer readable storage medium of any of claims 14-16, implementing the method further including:

calculating a new data field from data in two or more data fields in the primary database source; and

calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

18. The non-transitory computer readable storage medium of any of claims 14-17, implementing the method further comprising:

joining data from two or more objects in the primary database source; and

calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

19. The non-transitory computer readable storage medium of any of claims 14-18, implementing the method further comprising:

flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

AMENDED CLAIMS

received by the International Bureau on 26 september 2017 (26.09.2017)

1. A system for building a secure read-only database with data field level security inheritance, including:

extraction means for batch extraction of data fields comprising dimensions and measures from at least one primary database source;

compiling means for compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

security assignment means for assigning data field level security access to a plurality of the compiled data fields, the security means comprising selection means for a selection for a first compiled data field among the compiled data fields between security inheritance and security pinning, wherein

> selection of the security inheritance causes the security assignment means to assign data field level security to the first compiled data field by inheritance, from an extracted field in the primary database source; and
>
> selection of the security pinning causes the security assignment means to assign data field level security to the first compiled data field by security pinning of the first compiled data field to a reference field, with inheritance of the data field level security by the compiled data field from the pinned reference field;
>
> wherein the reference field is a second compiled data field extracted from the primary data source with data field level security inheritance;

building means for correlating the assigned data field level security with the compiled data fields during building of the read-only database.

2. The system of claim 1, with handling of unsecured source fields, further including:

the extraction means further for extraction of additional data fields from unsecured data sets that are not subject to data filed level security; and

security assignment means further for assigning field level security to the additional data fields with selection means for further selecting for a third data field among the additional data fields between explicit security and security pinning, wherein

> selection of the explicit security causes the security assignment means to assign data field level security by explicit specification of field level security for the third data

field, and

selection of the security pinning causes the security assignment means to assign data field level security to the third data field by security pinning of the third data field to the reference field in a reference table, with inheritance of the data field level security by the third data field from the pinned reference field.

3. The system of claim 2, the selection means further for selecting between explicit security and selection of explicit security causes the security assignment means to assign data field level security by explicit specification of field level security for the additional data fields.

4. The system of any of claims 1-3, wherein the read-only databases implement response times of under two seconds when searching over twenty million records and compiling aggregate statistics from selected records.

5. The system of any of claims 1-4, further including:

a new field means for calculating a new data field from data in two or more data fields in the primary database source; and

the security assignment means further for calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

6. The system of any of claims 1-5, further including:

a object joining means for joining data from two or more objects in the primary database source; and

the security assignment means further for calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

7. The system of any of claims 1-6, further including:

the selection means further for flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

8. A method of securing a read-only database with data field level security inheritance, including:

extracting data fields comprising dimensions and measures from at least one primary database source;

compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

assigning data field level security access to a plurality of the compiled data fields with selection between security inheritance and security pinning, including assignment of the field level security access to a first compiled data field among the compiled data fields, wherein

selection of the security inheritance causes assigning of data field level security to the first compiled data field by inheritance, from an extracted field in the primary database source; and

selection of the security pinning causes assigning of data field level security to the first compiled data field by security pinning of the first compiled data field to a reference field, with inheritance of the data field level security by the first compiled data field from the pinned reference field;

wherein the reference field is a second compiled data field extracted from the primary data source with data field level security inheritance; and

correlating the assigned data field level security with the compiled data fields during building of the read-only database.

9. The method of claim 8, with handling of unsecured source fields, further including:

extracting additional data fields from at least one unsecured data set that is not subject to data filed level security; and

assigning field level security to a third data field among the additional data fields by further selecting between explicit security and security pinning, wherein

selection of the explicit security causes assigning data field level security by explicit specification of field level security for the third data field, and

selection of the security pinning causes assigning data field level security to the third data field by security pinning of the third data field to the reference field in a reference table to the compiled data field, with inheritance of the data field level security by the third data field from the pinned reference field.

10. The method of claim 9, further including selecting between explicit security and selection of explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields.

11. The method of any of claims 8-10, further including:

calculating a new data field from data in two or more data fields in the primary database source; and

calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

12. The method of any of claims 8-11, further including:

joining data from two or more objects in the primary database source; and

calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

13. The method of any of claims 8-12, further including:

flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

14. A non-transitory computer readable storage medium impressed with computer program instructions to secure a read-only database, the instructions, when executed on a processor, implement a method comprising:

extracting data fields comprising dimensions and measures from at least one primary database source;

compiling the extracted data fields into at least one read-only database with immutable indexes for the dimensions among the extracted data fields;

assigning data field level security access to a plurality of the compiled data fields with selection between security inheritance and security pinning, including assignment of the field level security access to a first compiled data field among the compiled data fields, wherein

selection of the security inheritance causes assigning of data field level security to the first compiled data field by inheritance, from an extracted field in the primary database source; and

selection of the security pinning causes assigning of data field level security to the first compiled data field by security pinning of the first compiled data field to a reference field, with inheritance of the data field level security by the first compiled data field from the pinned reference field;

wherein the reference field is a second compiled data field extracted from the primary data source with data field level security inheritance; and

correlating the assigned data field level security with the compiled data fields during building of the read-only database.

15. The non-transitory computer readable storage medium of claim 14, with handling of unsecured source fields, implementing the method further including:

extracting additional data fields from at least one unsecured data set that is not subject to data filed level security; and

assigning field level security to a third data field among the additional data fields by further selecting between explicit security and security pinning, wherein

selection of the explicit security causes assigning data field level security by explicit specification of field level security for the third data field, and

selection of the security pinning causes assigning data field level security to the third data field by security pinning of the third data field to the reference field in a reference table to the compiled data field, with inheritance of the data field level security by the third data field from the pinned reference field.

16. The non-transitory computer readable storage medium of claim 15, implementing the method further including selecting between explicit security and selection of explicit security causes assigning data field level security by explicit specification of field level security for the additional data fields.

17. The non-transitory computer readable storage medium of any of claims 14-16, implementing the method further including:

calculating a new data field from data in two or more data fields in the primary database source; and

calculating data field level security in the new data field based on a combination of data field level security settings in the two or more data fields.

18. The non-transitory computer readable storage medium of any of claims 14-17, implementing the method further comprising:

joining data from two or more objects in the primary database source; and

calculating data field level security in one or more of data fields in the joined data based on a combination of data field level security settings in the two or more objects.

19. The non-transitory computer readable storage medium of any of claims 14-18, implementing the method further comprising:

flagging at least one of the data fields in the primary database source as a reference field for the security pinning and for causing display of a user interface that lists the flagged reference fields and enables user selection of the security pinning of the extracted fields to the reference fields.

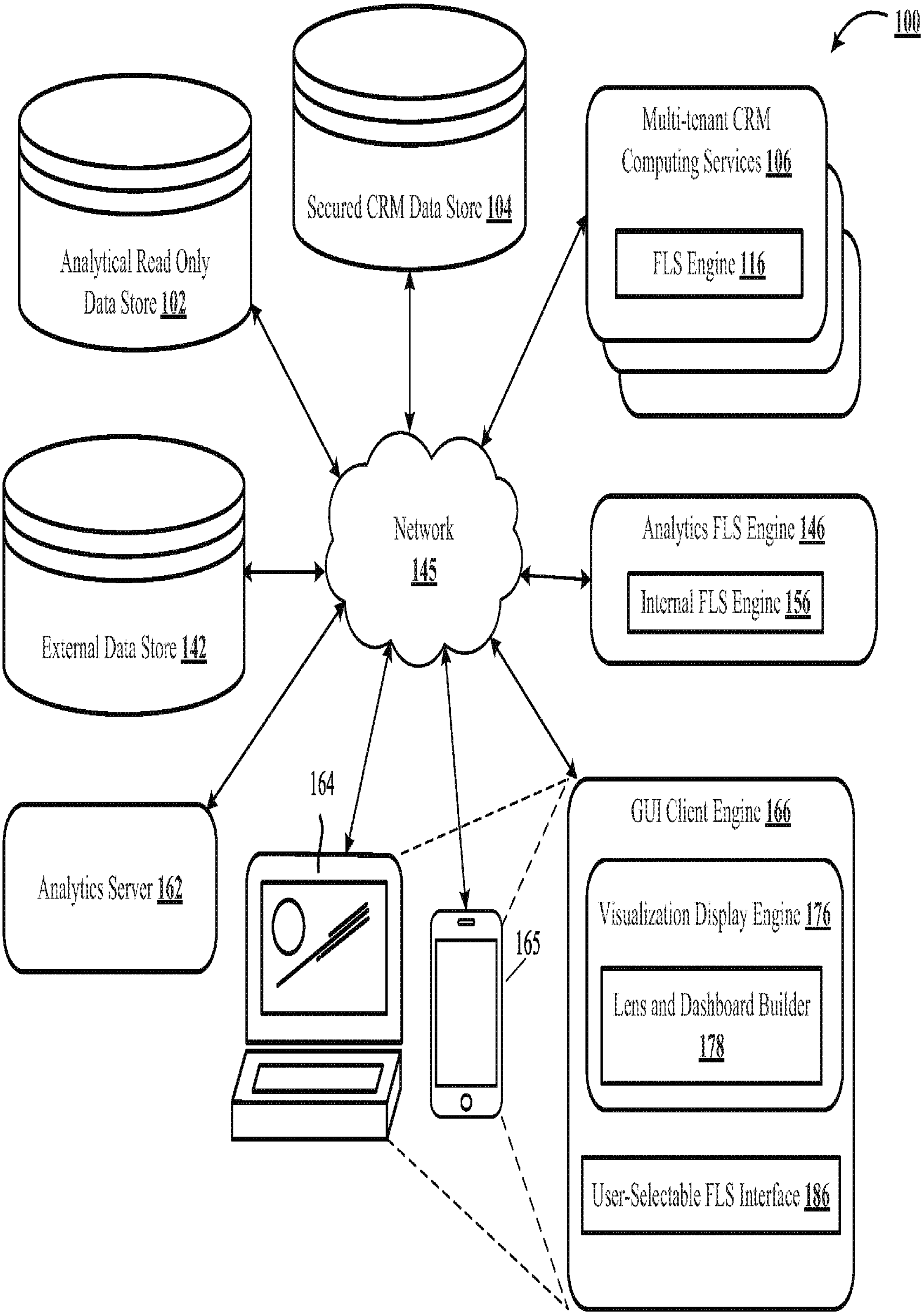

100
Analytical Read Only Data Store 102
Secured CRM Data Store 104
Multi-tenant CRM Computing Services 106
FLS Engine 116
External Data Store 142
Network 145
Analytics FLS Engine 146
Internal FLS Engine 156
Analytics Server 162
164
165
GUI Client Engine 166
Visualization Display Engine 176
Lens and Dashboard Builder 178
User-Selectable FLS Interface 186

FIG. 1

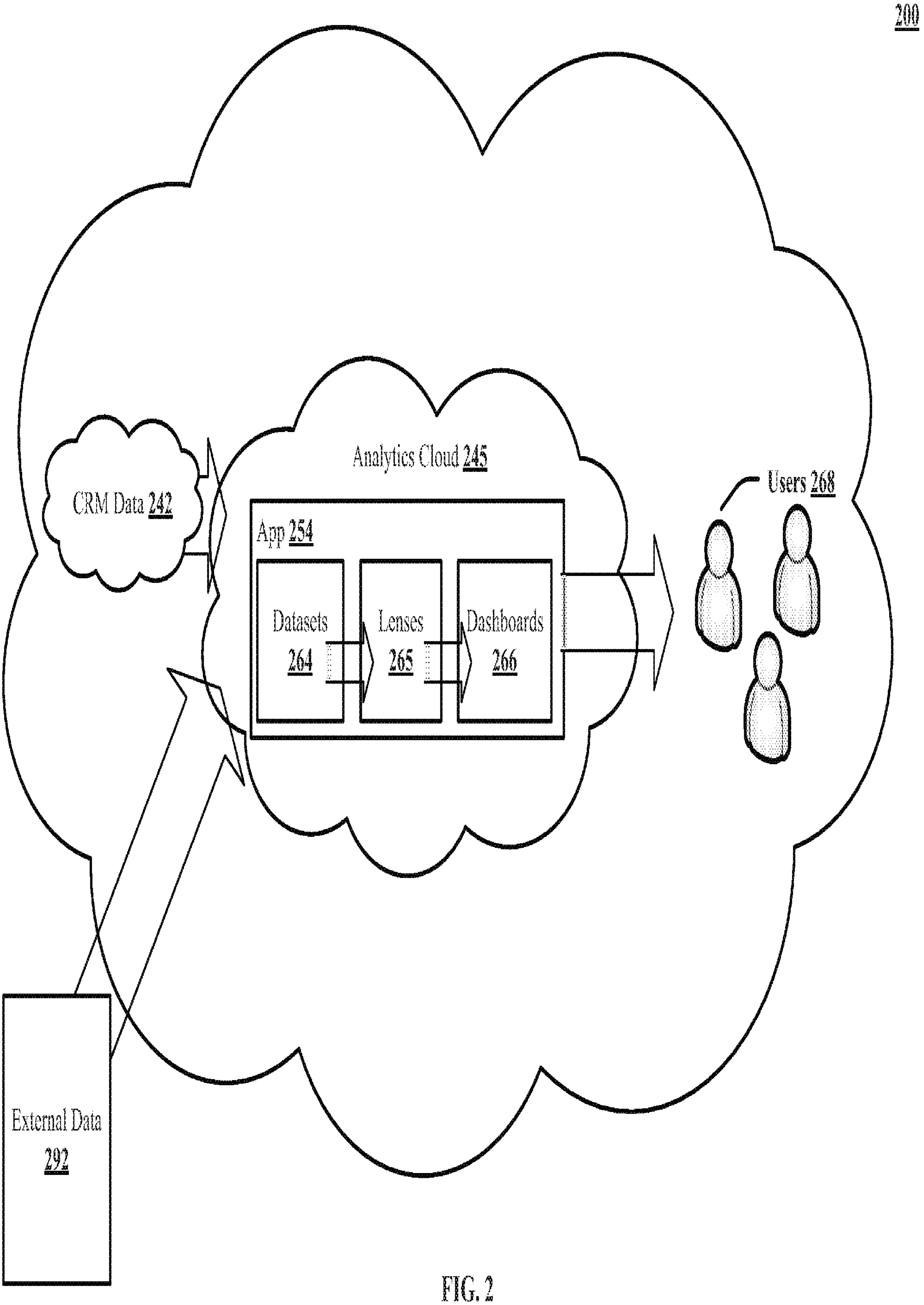

200
CRM Data 242
Analytics Cloud 245
App 254
Datasets 264
Lenses 265
Dashboards 266
Users 268
External Data 292

FIG. 2

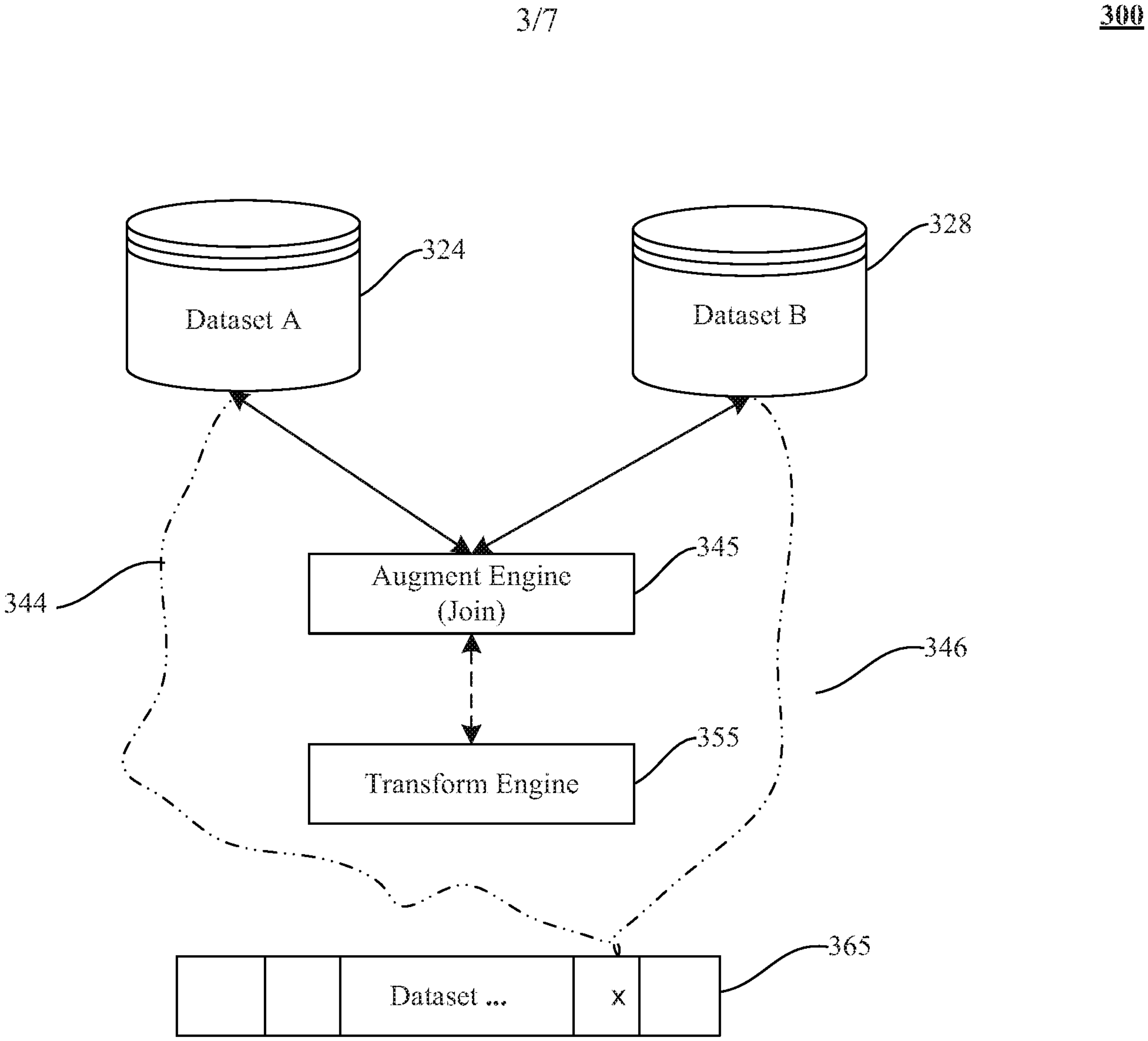

300
Dataset A
324
Dataset B
328
Augment Engine
(Join)
345
344
346
Transform Engine
355
Dataset ...
x
365

FIG. 3

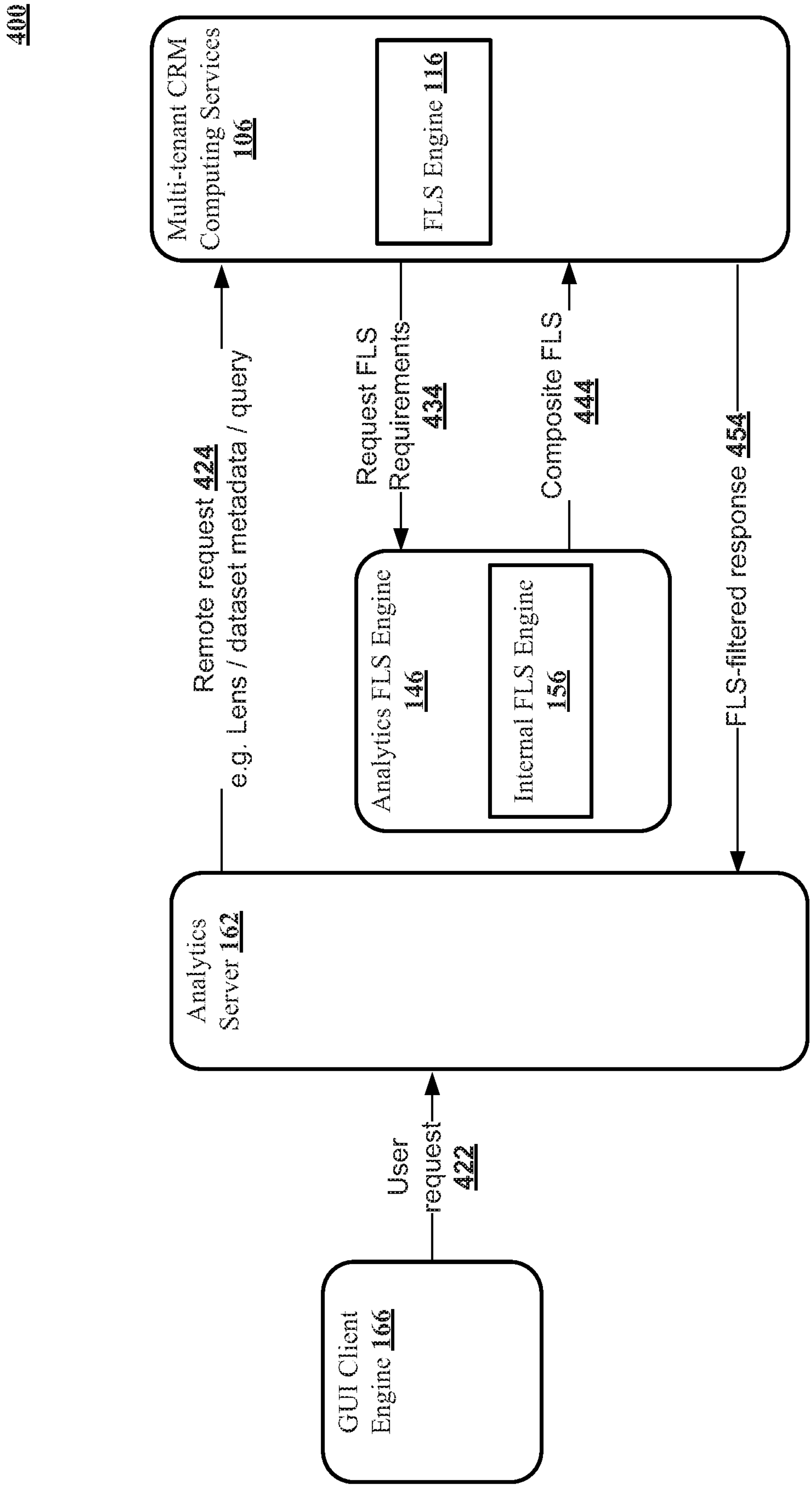
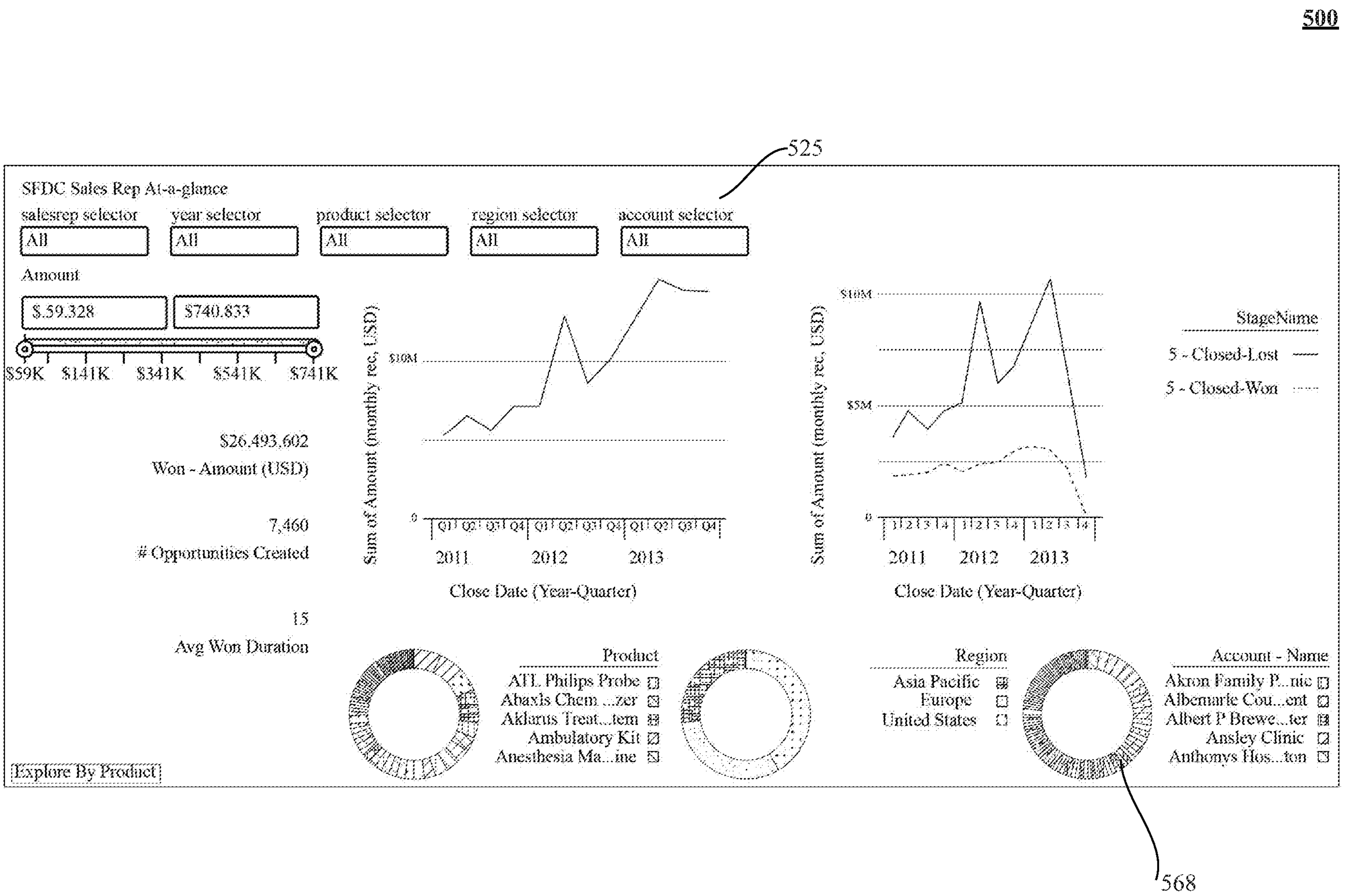

400
Multi-tenant CRM Computing Services 106
FLS Engine 116
Remote request 424 e.g. Lens / dataset metadata / query
Request FLS Requirements 434
Composite FLS 444
FLS-filtered response 454
Analytics FLS Engine 146
Internal FLS Engine 156
Analytics Server 162
User request 422
GUI Client Engine 166

FIG. 4



SFDC Sales Rep At-a-glance

salesrep selector: All
year selector: All
product selector: All
region selector: All
account selector: All

Amount: $.59.328 – $740.833

$26,493,602
Won - Amount (USD)

7,460
Opportunities Created

15
Avg Won Duration

Explore By Product

FIG. 5 Prior Art

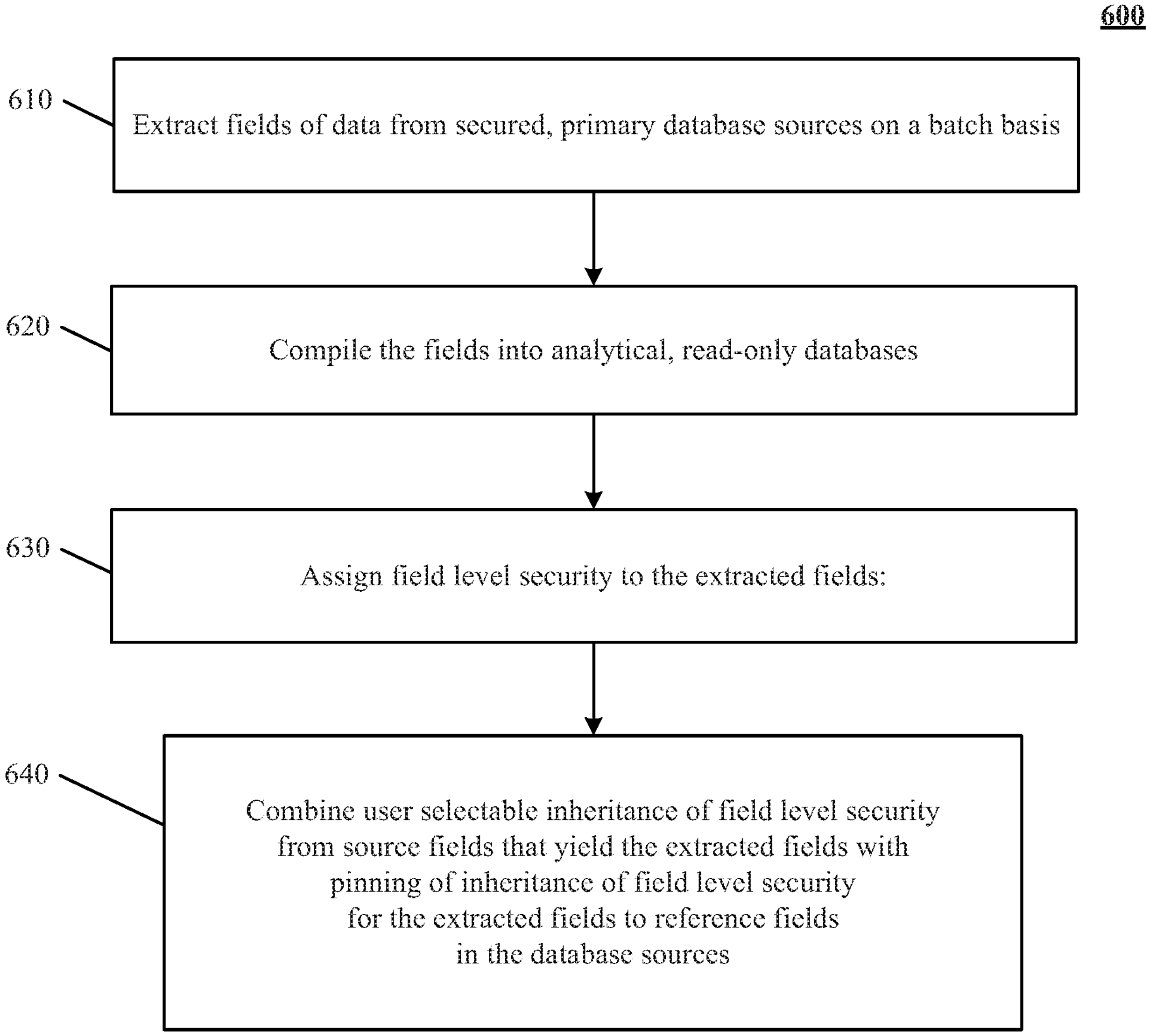

600
610
Extract fields of data from secured, primary database sources on a batch basis
620
Compile the fields into analytical, read-only databases
630
Assign field level security to the extracted fields:
640
Combine user selectable inheritance of field level security from source fields that yield the extracted fields with pinning of inheritance of field level security for the extracted fields to reference fields in the database sources

FIG. 6

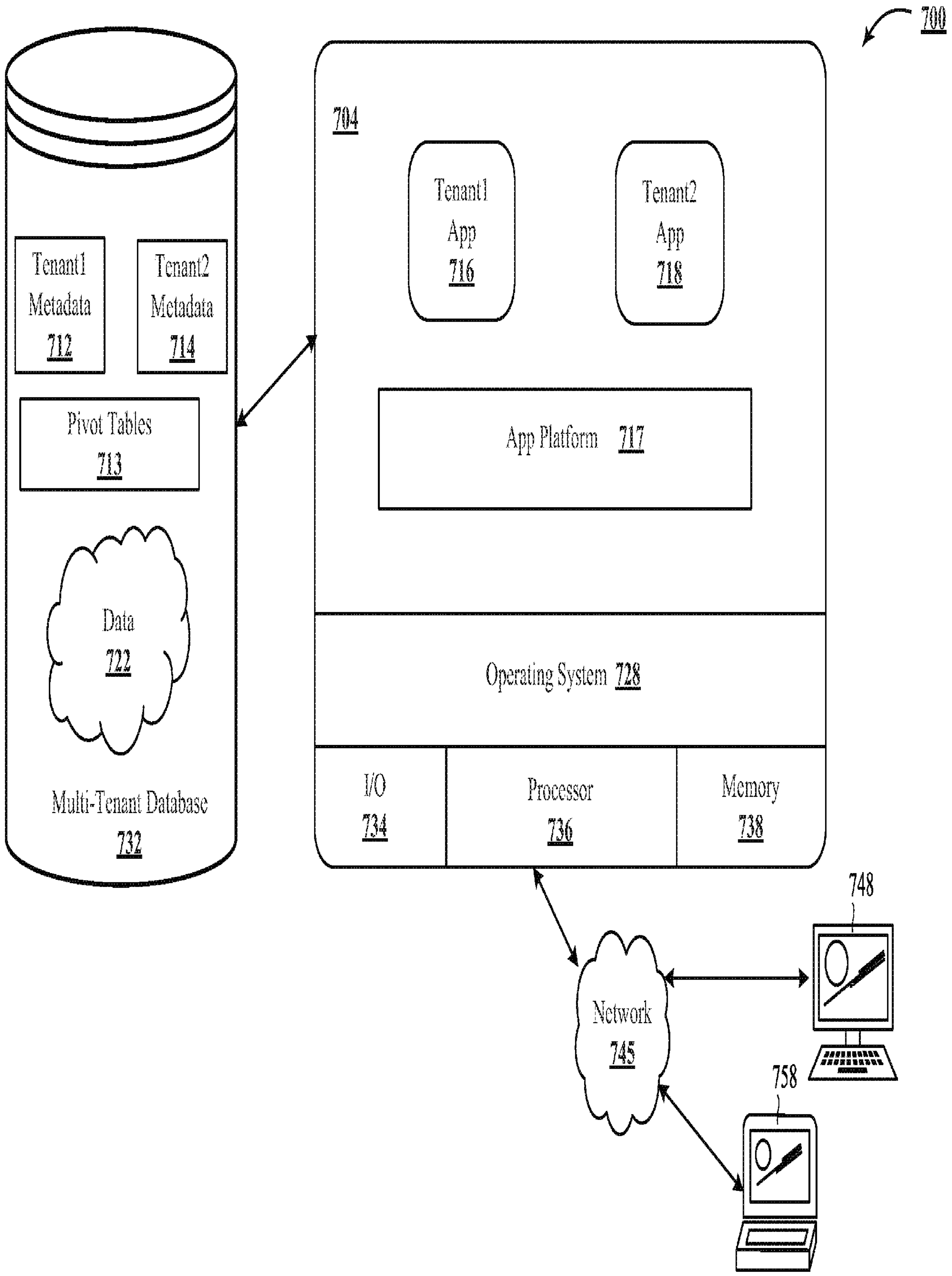

700
704
Tenant1 App 716
Tenant2 App 718
App Platform 717
Operating System 728
I/O 734
Processor 736
Memory 738
Tenant1 Metadata 712
Tenant2 Metadata 714
Pivot Tables 713
Data 722
Multi-Tenant Database 732
Network 745
748
758

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/027791

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

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

B. FIELDS SEARCHED

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

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2006/218157 A1 (SOUROV ALEXANDER A [US] ET AL) 28 September 2006 (2006-09-28) paragraphs [0004], [0008], [0027] - [0031], [0040], [0047] - [0049] -----	1-19
X	WO 2010/091456 A1 (TAP HOLDINGS PTY LTD [AU]; LEDWICH MARK JOSEPH [AU]; WILSON JAMES HENR) 19 August 2010 (2010-08-19) page 1, line 26 - page 5, line 28 page 16, line 6 - page 18, line 6 -----	1-19

□ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 July 2017	26/07/2017

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

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/027791

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2006218157 A1	28-09-2006	NONE	
WO 2010091456 A1	19-08-2010	AU 2010213346 A1	25-08-2011
		CA 2751383 A1	19-08-2010
		CN 102349050 A	08-02-2012
		EP 2396720 A1	21-12-2011
		US 2012005153 A1	05-01-2012
		WO 2010091456 A1	19-08-2010