September 2023

METHOD AND SYSTEM FOR CREATING A DATABASE 2 (Db2) CONNECTOR SUPPORT IN A DATAHUB

DEEPAK GARG
VISA

HARSIMRAN SINGH
VISA

JAYACHANDRA ADUSUMALLI
VISA

“METHOD AND SYSTEM FOR CREATING A DATABASE 2 (Db2) CONNECTOR SUPPORT IN A DATAHUB”

VISA

INVENTORS:

DEEPAK GARG

HARSIMRAN SINGH

JAYACHANDRA ADUSUMALLI

TECHNICAL FIELD

[0001] The present subject matter is, in general, related to data analysis techniques, and more particularly, but not exclusively to a method and a system for creating a database 2 (Db2) connector for a datahub.

BACKGROUND

[0002] Datahubs are becoming increasingly important as organizations move to a more data-driven approach. Datahub is an open-source extensible metadata platform, which enables data discovery, data observability, and federated governance. In general, the datahub is a centralized repository for storing and managing metadata. Metadata is a collection of data, and it includes information such as the name of the data source, the schema of the data, and the lineage of the data. For example, the datahub may be used to automatically generate data catalogues, create data lineage mappings, and identify data quality issues.

[0003] A Database 2 (Db2) is a collection of data management products to help users to handle big data. The "2" in Database 2 refers to a family of database management software, which shifted from a hierarchical to a relational database model. The Db2 is used to access data from both rational sources and non-rational data sources and in the cloud. For example, Db2 is used to query data from a MySQL database, or to query data from a MongoDB database. Currently Db2 is being used as one of the data-sources, however, the existing datahub configurations do not provide support to connect Db2 sources.

[0004] Currently, data from a variety of sources, including Db2, is being integrated using a data integration platform. For example, a Db2 crawler in a data integration platform is used to extract data from Db2 databases and load it into the data integration platform. However, the existing Db2 crawler is a simple crawler that does not provide many features. For example, it does not support crawling of nested tables or views. Additionally, it does not support crawling of data which is encrypted or compressed. Therefore, there is a necessity to create a Db2 crawler with an ingestion framework in datahub.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, explain the disclosed principles. In the figures, the left-most digit(s) of a reference number identifies the

figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of device or system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

[0006] **FIG. 1** is a schematic diagram illustrating an exemplary system which may be configured for creating a database 2 (Db2) connector for a datahub, in accordance with some embodiments of the present disclosure.

[0007] **FIG. 2** illustrates an exemplary flow diagram of a method for creating a database 2 (Db2) connector for a datahub, in accordance with some embodiments of the present disclosure.

[0008] **FIG. 3** shows a block diagram illustrating an exemplary computer system for implementing embodiments consistent with the present disclosure.

[0009] The figures depict embodiments of the disclosure for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the disclosure described herein.

DESCRIPTION OF THE DISCLOSURE

[0010] It is to be understood that the present disclosure may assume various alternative variations and step sequences, except where expressly specified to the contrary. It is also to be understood that the specific devices and processes illustrated in the attached drawings and described in the following specification are simply exemplary and non-limiting embodiments or aspects. Hence, specific dimensions and other physical characteristics related to the embodiments or aspects disclosed herein are not to be considered as limiting.

[0011] In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

[0012] While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the

disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

[0013] The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device, or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

[0014] The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

[0015] The terms "including", "comprising", “having” and variations thereof mean "including but not limited to" unless expressly specified otherwise.

[0016] As used herein, the terms “communication” and “communicate” may refer to the reception, receipt, transmission, transfer, provision, and/or the like of information (e.g., data, signals, messages, instructions, commands, and/or the like). For one unit (e.g., a device, a system, a component of a device or system, combinations thereof, and/or the like) to be in communication with another unit means that the one unit can receive information directly or indirectly from and/or transmit information to the other unit. This may refer to a direct or indirect connection (e.g., a direct communication connection, an indirect communication connection, and/or the like) that is wired and/or wireless in nature. Additionally, two units may be in communication with each other even though the information transmitted may be modified, processed, relayed, and/or routed between the first and second unit. For example, a first unit may be in communication with a second unit even though the first unit passively receives information and does not actively transmit information to the second unit. As another example, a first unit may be in communication with a second unit if at least one intermediary unit (e.g., a third unit located between the first unit and the second unit) processes information received from the first unit and communicates the processed information to the second unit. In

some non-limiting embodiments, a message may refer to a network packet (e.g., a data packet and/or the like) that includes data. It will be appreciated that numerous other arrangements are possible.

[0017] As used herein, the term “computer” may refer to any computing device that includes the necessary components to receive, process, and output data, and normally includes a display, a processor, a memory, an input device, and a network interface. A “computing system” may include one or more computing devices or computers.

[0018] **FIG. 1** illustrates an exemplary environment 100 of a data ingestion system 103, which is configured to facilitate the collection, import, and storage of data from various sources into a centralized repository, such as a datahub. The datahub is a metadata management platform which helps catalog, organize, and provide insights about data assets. In an embodiment, environment 100 may additionally comprise, without limitation, one or more data sources 101_N, the data ingestion system 103, a metadata change proposal 105 and one or more data sinks 107_N. The data sources 101_N and the system 103 may be connected via a predefined communication network (not shown in FIG. 1). Such a communication network may include, without limitation, a direct interconnection, a Local Area Network (LAN), a Wide Area Network (WAN), a wireless network (e.g., using Wireless Application Protocol), the Internet, and the like.

[0019] In an embodiment, a Database (Db2) database 109 contains various types of data such as tables, records, and schemas. The Db2 database 109 is a specific type of Db2 source. A Db2 connector is configured to extract data from the Db2 database 109 (not shown in FIG. 1). The data sources 101_N may be utilized to collect various data from the Db2 database 109 via the Db2 connector. Thereafter, in the ingestion process, system 103 receives the Db2 data sources 101_N. In an embodiment, Db2 data sources 101_N are the sources which may be accessed by Db2 109. The Db2 data sources 101_N may be either relational databases or non-relational databases.

[0020] In an implementation, the data ingestion system 103 may include a set of processing tools used to collect data from various data sources 101_N. In other words, the system 103 may be used to ingest data from a variety of data sources 101_N. The real-time data ingestion may require continuous capture and immediate procession of data as it becomes available. For example, Apache Kafka® is used for real-time ingestion. The system 103 may be written in

Python programming language and allow adding custom metadata sources. In some embodiments, the system 103 may include one or more processors, an I/O interface, and a memory. In some embodiments, the memory of the system 103 may be communicatively coupled to the one or more processors of the system 103. The system 103 may be implemented in a variety of computing systems, such as a laptop computer, a desktop computer, a Personal Computer (PC), a notebook, a smartphone, a tablet, e-book readers, a server, a network server, a cloud-based server and the like. In an embodiment, the system 103 may be a dedicated server or may be a cloud-based server.

[0021] In an embodiment, the metadata change proposal 105 is utilized to establish a structured and controlled process for managing metadata modifications. The metadata change proposal 105 identifies a need for a change in metadata based on the request received from the system 103. Once the change required is identified, the metadata change proposal is created. As an example, the proposal includes details related to the proposed change, for example, updating or changing data source information, the affected metadata records and so on. Further, the metadata changes proposal is transmitted to various data sinks 107_N, such as REST API and Kafka. As a result, the metadata change proposal 105 may be used to ensure that data sources 101_N and data sinks 107_N are always up to date.

[0022] In an embodiment, the data sinks 107_N are destinations for the metadata. That is, after configuring ingestion for datahub, the metadata is sent to the datahub over either the REST API (datahub-REST) or the Kafka sink (datahub-Kafka).

[0023] Consider an example, where a user wishes to change the schema or tables or catalogs of a data source. Here, a Db2 connector is configured for a datahub to support the data ingestion system 103. A metadata change proposal 105 may be generated to notify the system 103 of the change. The system 103 continuously monitors the metadata associated with data sources 101_N and Db2 database 109. Thereafter, the system 103 updates the data catalogs to reflect the new schema. The update may involve modifying the metadata records and data information or data schema definitions. As a result, creating the Db2 connector with an ingestion framework in the datahub may expose one or more features provided by the datahub, such as transformations of data before ingestion and profiling and reporting ingestion pipelines.

[0024] **FIG. 2** illustrates an exemplary flow diagram of a method for creating a database 2 (Db2) connector for a datahub, in accordance with some embodiments of the present disclosure.

[0025] In an embodiment, at block 201, the method comprises obtaining the input data sources 101_N from a Database 2 (Db2) database 109. The sources 101_N pull metadata from a variety of data systems. As an example, these sources 101_N may be created using Python libraries, which gives features such as connection to Db2, queries to fetch metadata, and the like. Thereafter, in block 203, the method comprises ingesting, by a data ingestion system 103, the obtained data sources 101_N, that is, Db2 sources in a datahub. In other words, a Db2 connector is configured in a datahub environment, wherein the Db2 connector establishes a connection to the Db2 sources database and pulls data changes from the Db2 database 109. Subsequently, in block 205, the method comprises generating metadata change proposal 105 requests based on the Db2 sources. Finally, as indicated in block 207, the generated metadata change proposal is sent to the datahub through the data sinks 107_N, wherein the data sinks 107_N include a datahub over REST API and a datahub over Kafka. The data sinks 107_N are primarily utilized for moving the obtained metadata into the datahub. The REST API and Kafka are configured in the datahub to accept metadata changes, along with the process and validating the incoming metadata change messages. As a result, configuring the Db2 connector may effectively capture changes from the Db2 sources 101_N, propose metadata changes, and send it to the datahub, either over REST or Kafka, thereby enhancing metadata management processes.

Advantages of the proposed solution:

[0026] In an embodiment, the present disclosure enhances data lineage tracking and metadata management with datahub. That is, the present disclosure enables a data integration platform to capture critical information about data sources, and transformations, while ensuring data tracking is accurately documented.

[0027] In an embodiment, the present disclosure allows the datahub to directly access the data from Db2 sources, thereby simplifying the process of utilizing data assets.

[0028] In an embodiment, the present disclosure provides a centralized catalog which lists all available Db2 datasets combined with the Db2.

[0029] In an embodiment, the present disclosure helps in identifying data quality issues and tracking the quality of data over a period of time.

General computer system:

[0030] **FIG. 3** illustrates a block diagram of an exemplary computer system for implementing embodiments consistent with the present disclosure.

[0031] In an embodiment, **FIG. 3** illustrates a block diagram of an exemplary computer system 300 which may be used to implement the data ingestion system 103 for creating a database 2 (Db2) connector in a datahub. In an embodiment, the computer system 300 may include a central processing unit ("CPU" or "processor") 302. The processor 302 may include at least one data processor for executing processes in Virtual Storage Area Network. Processor 302 may include at least one data processor for executing program components for executing user or system-generated business processes. The processor 302 may include specialized processing units such as integrated system (bus) controllers, memory management control units, floating point units, graphics processing units, digital signal processing units, etc.

[0032] The processor 302 may be disposed in communication with one or more Input/Output (I/O) devices (312 and 313) via I/O interface 301. The I/O interface 301 employ communication protocols/methods such as, without limitation, audio, analog, digital, monoaural, Radio Corporation of America (RCA) connector, stereo, IEEE-1394 high-speed serial bus, serial bus, Universal Serial Bus (USB), infrared, Personal System/2 (PS/2) port, Bayonet Neill-Concelman (BNC) connector, coaxial, component, composite, Digital Visual Interface (DVI), High-Definition Multimedia Interface (HDMI), Radio Frequency (RF) antennas, S-Video, Video Graphics Array (VGA), IEEE 802.11b/g/n/x, Bluetooth, cellular, for example, Code-Division Multiple Access (CDMA), High-Speed Packet Access (HSPA+), Global System for Mobile communications (GSM), Long-Term Evolution (LTE), Worldwide Interoperability for Microwave access (WiMax), or the like, etc.

[0033] Using the I/O interface 301, the computer system 300 may communicate with one or more I/O devices such as input devices 312 and output devices 313. For example, the input devices 312 may be an antenna, keyboard, mouse, joystick, (infrared) remote control, camera, card reader, fax machine, dongle, biometric reader, microphone, touch screen, touchpad, trackball, stylus, scanner, storage device, transceiver, video device/source, etc. The output devices 313 may be a printer, fax machine, video display (e.g., Cathode Ray Tube (CRT), Liquid Crystal Display (LCD), Light-Emitting Diode (LED), plasma, Plasma Display Panel (PDP), Organic Light-Emitting Diode display (OLED) or the like), audio speaker, etc.

[0034] In some embodiments, the processor 302 may be disposed in communication with a communication network 309 via a network interface 303. The network interface 303 may communicate with the communication network 309. The network interface 303 may employ connection protocols including, without limitation, direct connect, ethernet (e.g., twisted pair 10/100/1000 Base T), Transmission Control Protocol/Internet Protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc. The communication network 309 may include, without limitation, a direct interconnection, Local Area Network (LAN), Wide Area Network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, etc. Using the network interface 303 and the communication network 309, the computer system 300 may communicate with a database 314, which may be the enrolled templates database 313. The network interface 303 may employ connection protocols include, but not limited to, direct connect, ethernet (e.g., twisted pair 10/100/1000 Base T), Transmission Control Protocol/Internet Protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc.

[0035] In some embodiments, the communication network 309 includes, but is not limited to, a direct interconnection, a Peer-to-Peer (P2P) network, Local Area Network (LAN), Wide Area Network (WAN), wireless network (for example, using Wireless Application Protocol), the Internet, Wi-Fi, and such. The communication network 309 may either be a dedicated network or a shared network, which represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP), etc., to communicate with each other. Further, the communication network 309 may include a variety of network devices, including routers, bridges, servers, computing devices, storage devices, etc.

[0036] In some embodiments, the processor 302 may be disposed of in communication with a memory 305 (e.g., RAM, ROM, etc. not shown in **FIG. 3**) via a storage interface 304. The storage interface 304 may connect to memory 305 including, without limitation, memory drives, removable disc drives, etc., employing connection protocols such as, Serial Advanced Technology Attachment (SATA), Integrated Drive Electronics (IDE), IEEE-1394, Universal Serial Bus (USB), fiber channel, Small Computer Systems Interface (SCSI), etc. The memory drives may further include a drum, magnetic disc drive, magneto-optical drive, optical drive, Redundant Array of Independent Discs (RAID), solid-state memory devices, solid-state drives, etc.

[0037] In some embodiments, the memory 305 may store a collection of program or database components, including, without limitation, user interface 306, an operating system 307, a web browser 308 etc. In some embodiments, computer system 300 may store user/application data, such as, the data, variables, records, etc., as described in this disclosure. Such databases may be implemented as fault-tolerant, relational, scalable, secure databases such as Oracle or Sybase.

[0038] In some embodiments, the operating system 307 may facilitate resource management and operation of the computer system 300. Examples of operating systems include, without limitation, Apple Macintosh OS X™, UNIX™, Unix-like system distributions (e.g., Berkeley Software Distribution (BSD), FreeBSD, Net BSD™, Open BSD™, etc.), Linux distributions (e.g., Red Hat, Ubuntu, K-Ubuntu, etc.), International Business Machines (IBM™) OS/2™, Microsoft Windows (XP™, Vista/7/8, etc.), Apple iOS, Google Android, BlackBerry operating system (OS), or the like. The User Interface 306 may facilitate display, execution, interaction, manipulation, or operation of program components through textual or graphical facilities. For example, user interfaces may provide computer interaction interface elements on a display system operatively connected to the computer system 300, such as cursors, icons, checkboxes, menus, scrollers, windows, widgets, etc. Graphical User Interfaces (GUIs) may be employed, including, without limitation, Apple® Macintosh® operating systems' Aqua®, IBM® OS/2®, Microsoft® Windows® (e.g., Aero, Metro, etc.), web interface libraries (e.g., ActiveX®, Java®, JavaScript®, AJAX, HTML, Adobe® Flash®, etc.), or the like.

[0039] In some embodiments, the computer system 300 may implement web browser 308 stored program components. Web browser 308 may be a hypertext viewing application, such as Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Apple Safari, etc. Secure web browsing may be provided using secure hypertext transport protocol (HTTPS), Secure Sockets Layer (SSL), Transport Layer Security (TLS), etc. Web browsers 308 may utilize facilities such as AJAX, DHTML, Adobe Flash, JavaScript, Application Programming Interfaces (APIs), etc.

[0040] In some embodiments, the computer system 300 may implement a mail server stored program component. The mail server may be an Internet mail server such as Microsoft Exchange, or the like. The mail server may utilize facilities such as ASP, ActiveX, ANSI C++/C#, Microsoft .NET, Common Gateway Interface (CGI) scripts, Java, JavaScript, PERL, PHP, Python, WebObjects, etc. The mail server may utilize communication protocols such as

Internet Message Access Protocol (IMAP), Messaging Application Programming Interface (MAPI), Microsoft Exchange, Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), or the like.

[0041] In some embodiments, the computer system 300 may implement a mail client stored program component. The mail client may be a mail viewing application, such as APPLE® MAIL, MICROSOFT® ENTOURAGE®, MICROSOFT® OUTLOOK®, MOZILLA® THUNDERBIRD®, etc.

[0042] Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer-readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term "computer-readable medium" should be understood to include tangible items and exclude carrier waves and transient signals, i.e., be non-transitory. Examples include Random Access Memory (RAM), Read-Only Memory (ROM), volatile memory, non-volatile memory, hard drives, Compact Disc (CD) ROMs, DVDs, flash drives, disks, and any other known physical storage media.

[0043] The described operations may be implemented as a method, system or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The described operations may be implemented as code maintained in a "non-transitory computer-readable medium", where a processor may read and execute the code from the computer-readable medium. The processor is at least one of a microprocessor and a processor capable of processing and executing the queries. A non-transitory computer-readable medium may include media such as magnetic storage medium (e.g., hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, DVDs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, Flash Memory, firmware, programmable logic, etc.), etc. Further, non-transitory computer-readable media may include all computer-readable media except for transitory. The code implementing the described operations may further be implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.).

[0044] The illustrated steps are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments. Also, the words "comprising," "having," "containing," and "including," and other similar forms are intended to be equivalent in meaning and be open ended in that an item or items following any one of these words is not meant to be an exhaustive listing of such item or items or meant to be limited to only the listed item or items. It must also be noted that as used herein, the singular forms “a,” “an,” and “the” include plural references unless the context clearly dictates otherwise.

[0045] Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. The term “computer-readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., are non-transitory. Examples include Random Access Memory (RAM), Read-Only Memory (ROM), volatile memory, non-volatile memory, hard drives, CD ROMs, DVDs, flash drives, disks, and any other known physical storage media.

[0046] Finally, the language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the embodiments of the disclosure is intended to be illustrative, but not limiting, of the scope of the disclosure.

[0047] With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

“METHOD AND SYSTEM FOR CREATING A DATABASE 2 (Db2) CONNECTOR SUPPORT IN A DATAHUB”

ABSTRACT

The present disclosure relates to a method and a system for creating a Database 2 (Db2) connector support for a datahub. The present disclosure suggests obtaining the input data sources from a Db2 database. Thereafter, the obtained data sources are ingested into the datahub. Subsequently, a metadata change proposal is generated based on the Db2 sources. Further, the generated metadata change proposal is sent to the datahub though the data sinks, wherein the data sinks include REST API and Kafka.

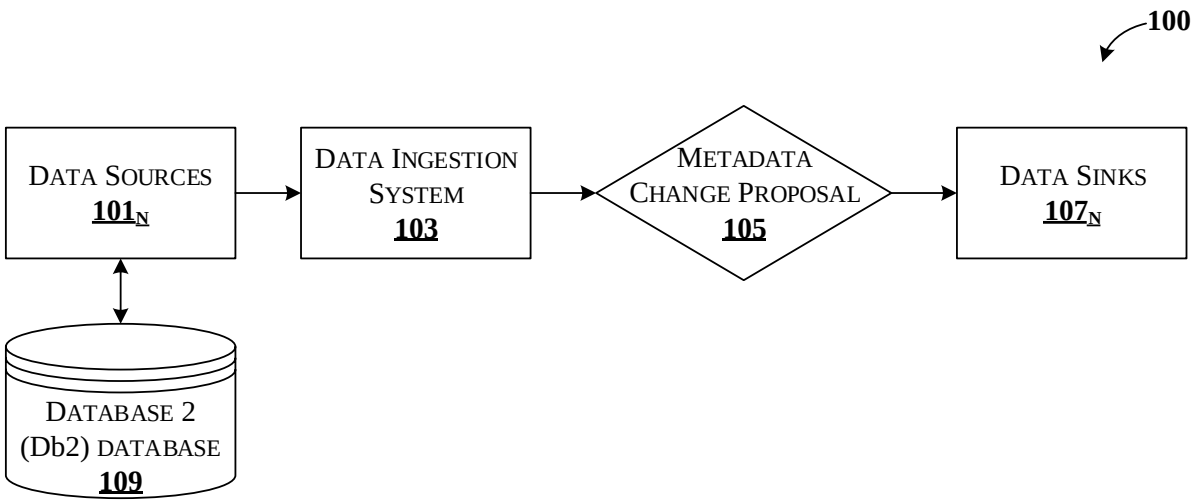
100
DATA SOURCES
101N
DATA INGESTION SYSTEM
103
METADATA CHANGE PROPOSAL
105
DATA SINKS
107N
DATABASE 2 (Db2) DATABASE
109

FIG. 1

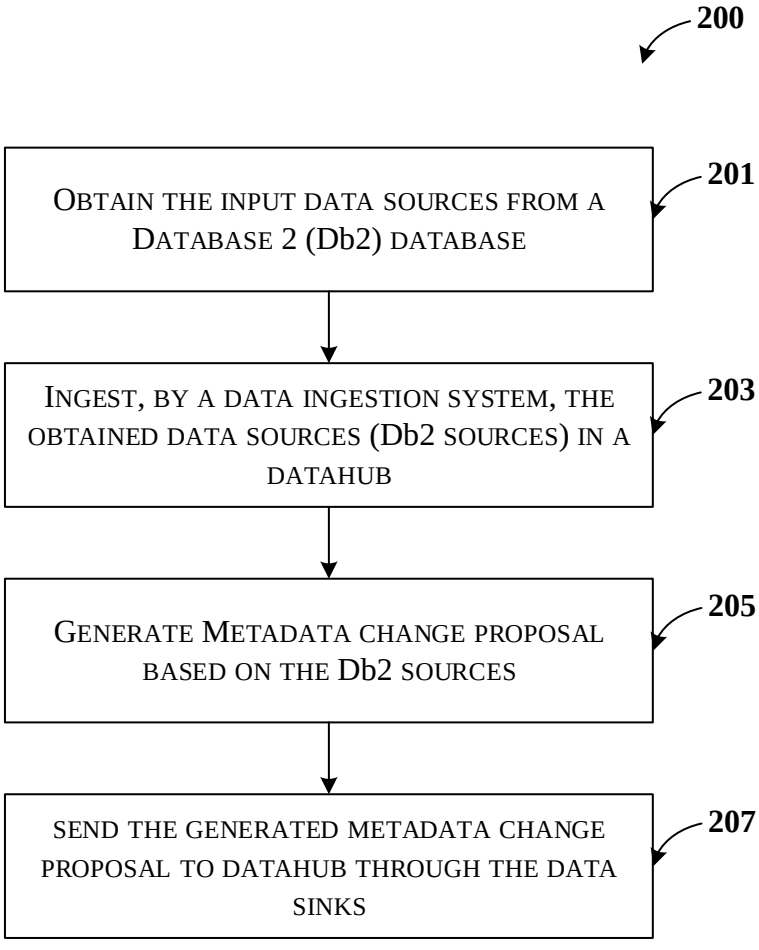
200
OBTAIN THE INPUT DATA SOURCES FROM A DATABASE 2 (Db2) DATABASE
201
INGEST, BY A DATA INGESTION SYSTEM, THE OBTAINED DATA SOURCES (Db2 SOURCES) IN A DATAHUB
203
GENERATE METADATA CHANGE PROPOSAL BASED ON THE Db2 SOURCES
205
SEND THE GENERATED METADATA CHANGE PROPOSAL TO DATAHUB THROUGH THE DATA SINKS
207

FIG. 2

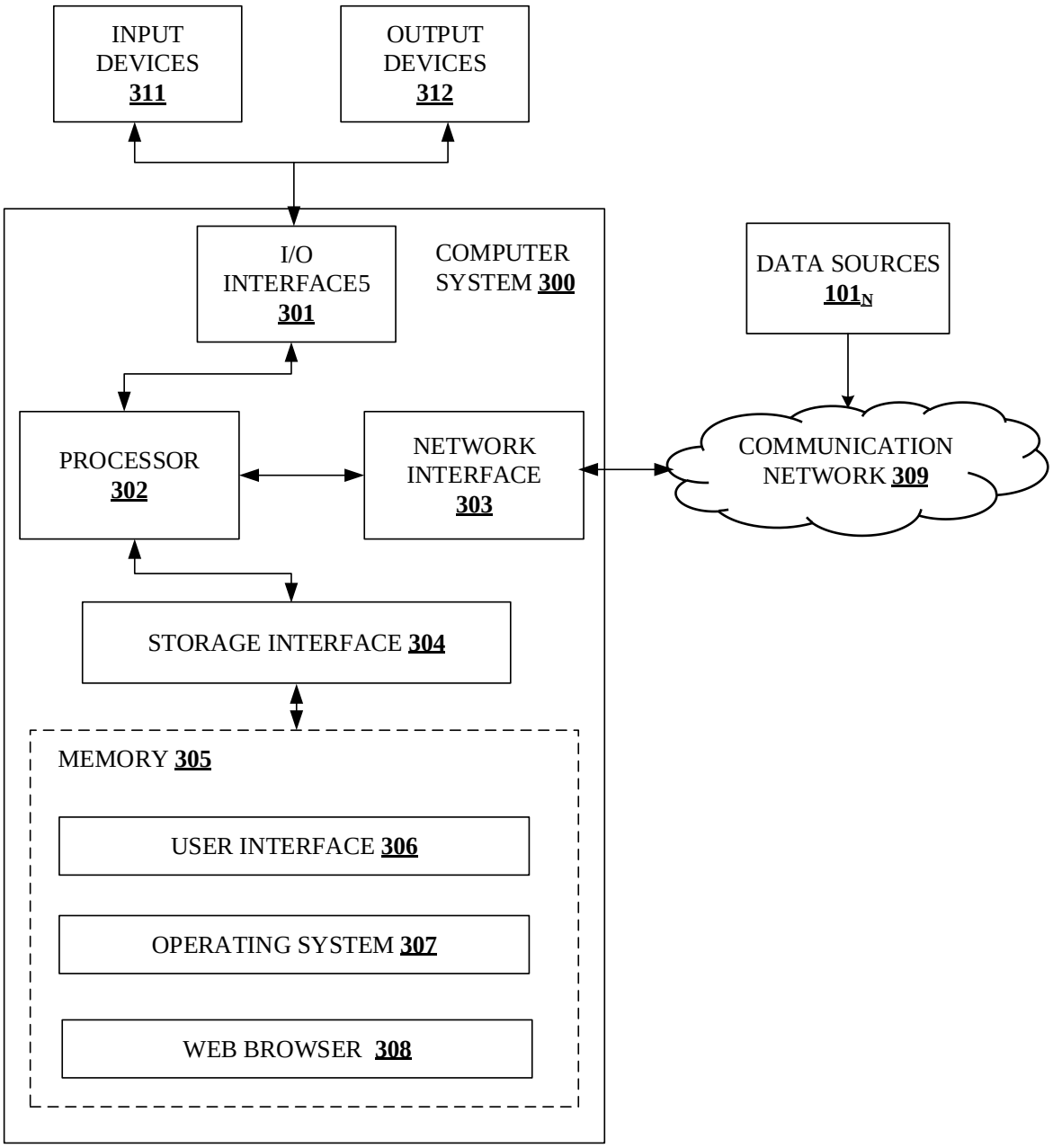
INPUT DEVICES 311
OUTPUT DEVICES 312
I/O INTERFACE5 301
COMPUTER SYSTEM 300
DATA SOURCES 101N
PROCESSOR 302
NETWORK INTERFACE 303
COMMUNICATION NETWORK 309
STORAGE INTERFACE 304
MEMORY 305
USER INTERFACE 306
OPERATING SYSTEM 307
WEB BROWSER 308

FIG. 3