

DATA PROCESSING METHOD AND DEVICE

TECHNICAL FIELD

[0001] This specification relates to the field of big data technologies, and in particular, to a data processing method and a data processing device.

5

BACKGROUND

[0002] In recent years, more enterprises start to seek development overseas. A data privacy compliance risk gradually becomes a major challenge that various multi-national enterprises have to face.

[0003] Most multi-national enterprises, in consideration of both development costs and operation and maintenance costs, will choose to store service data from multiple entities (e.g. multiple countries) in one data station. However, data peeling, data management and control, or data filing needs to be able to be quickly performed on data of different data holding entities, to respond to complex geopolitical changes and regional regulatory compliance challenges. Because different countries have different management and control requirements in terms of data integration, storage, flow, calculation, etc., a data holding entity identification capability is of great significance to compliant operation of an international service.

[0004] In a possible technical solution, different data containers can be built on a public cloud, to respectively store data of different data holding entities, for example, different countries. However, limited by costs, data of different data holding entities sometimes may be stored in one or more data center stations. Therefore, in the data center stations, data of a plurality of data holding entities is still stored in a hybrid manner, and it is difficult to identify data holding entities of the hybrid data.

SUMMARY

[0005] This specification mainly aims to provide a data processing method and device, so that data holding entities of hybrid data from multiple entities can be identified.

[0006] According to a first aspect, this specification provides a data processing method, including: determining at least one table field of a to-be-labeled table and a plurality of seed dimension tables, wherein the seed dimension table includes an entity association field set including a plurality of entity association fields, and wherein each of the plurality of entity association fields is used to associate an entity of the to-be-labeled table with an entity of the plurality of seed dimension tables; if an entity association field corresponding to the at least one table field exists in an entity association field set of at least one seed dimension table, determining labeling information corresponding to data of each row in the to-be-labeled table from the at least one seed dimension table based on the corresponding entity association field; and labeling data of each row in the to-be-labeled table based on the labeling information, where the seed dimension table records the labeling information of data of a plurality of rows corresponding to the entity association field, and the labeling information includes information on data holding entity.

[0007] According to a second aspect, this specification provides a data processing device, including: at least one storage medium, including at least one instruction set; and at least one processor, communicatively connected to the at least one storage medium. When the device runs, the at least one processor reads the at least one instruction set, and executes the method according to the first aspect based on an indication of the at least one instruction set.

[0008] According to a third aspect, this specification provides a non-transitory storage medium. The non-transitory storage medium stores at least one group of executable instructions, and when the at least one group of executable instructions are executed by at least one processor, the method according to the first aspect is performed.

[0009] It can be learned from the technical solutions that according to the data processing method and the data processing device provided in this specification, the entity association field set of the seed dimension table is preset, so that a plurality of entity association fields of a service entity of the seed dimension table can be obtained. In addition, the labeling information corresponding to data of each row in the to-be-labeled table is determined based on the entity association field that is of the seed dimension table and that corresponds to the table field of the to-be-labeled table, so that a field that is of the to-be-labeled table and that is associated with the service entity of the seed dimension table can be accurately identified, to perform accurate data holding entity identification on the to-be-labeled table. In addition, data of each row in the to-be-labeled table is labeled based on the labeling information that includes information on the data

holding entity and that is recorded in the seed dimension table, to automatically perform data holding entity labeling on data of each row in the to-be-labeled table, so that entity identification can be performed on hybrid data from multiple entities (e.g. multiple countries).

[0010] Other functions of the data processing method and device provided in this specification are partially listed below in this specification. Based on the descriptions, content described by using the following figures and examples is apparent to a person of ordinary skill in the art. Creative aspects of the data processing method and system provided in this specification can be fully explained through practice or by using the method, apparatus, and combination described in the following detailed examples.

BRIEF DESCRIPTION OF DRAWINGS

[0011] To describe the technical solutions in the embodiments of this specification more clearly, the accompanying drawings required for describing the embodiments are described briefly below. Clearly, the accompanying drawings in the following descriptions show some embodiments of this specification, and a person of ordinary skill in the art can still derive other accompanying drawings from these accompanying drawings without creative efforts.

[0012] FIG. 1 is a schematic diagram of an implementation environment of a data processing method according to an embodiment of this specification;

[0013] FIG. 2 is a schematic structural diagram of a data processing device according to some embodiments of this specification;

[0014] FIG. 3 is a schematic flowchart of a data processing method according to some embodiments of this specification;

[0015] FIG. 4 is a schematic flowchart of determining labeling information of a to-be-labeled table according to some embodiments of this specification;

[0016] FIG. 5 is a schematic flowchart of determining a lineage field according to some embodiments of this specification;

[0017] FIG. 6 is a diagram of a functional architecture of a data processing device according to some embodiments of this specification;

[0018] FIG. 7 is a schematic flowchart of a data processing method according to some other embodiments of this specification;

[0019] FIG. 8 is a schematic flowchart of a data processing method according to still some other embodiments of this specification; and

[0020] FIG. 9 is a schematic diagram of an entity model tree according to yet some other embodiments of this specification.

5

DESCRIPTION OF EMBODIMENTS

[0021] The following descriptions provide specific application scenarios and requirements of this specification, so that a person skilled in the art can manufacture and use content of this specification. For the person skilled in the art, various local modifications to the disclosed embodiments are apparent, and the general principles defined herein can be applied to other
10 embodiments and applications without departing from the spirit and scope of this specification. Therefore, this specification is not limited to the shown embodiments, but has a widest scope consistent with that of the claims.

[0022] The terms used herein are only intended to describe a particular example embodiment, but do not impose a limitation. For example, unless otherwise specified in the context, the singular
15 forms "one", "a", and "the" used herein can also include a plural form. When used in this specification, the terms "include", "comprise", and/or "have" indicate/indicates existence of an associated integer, step, operation, element, and/or component, but does not exclude existence of one or more other features, integers, steps, operations, elements, components, and/or groups, or another feature, integer, step, operation, element, component, and/or group can be added to the
20 system/method.

[0023] The flowchart used in this specification illustrates operations implemented by a system in some embodiments of this specification. It should be clearly understood that operations in the flowchart can be implemented out of sequence. On the contrary, operations can be implemented reversely or simultaneously. In addition, one or more other operations can be added to the
25 flowchart. One or more operations can be removed from the flowchart.

[0024] First, noun terms related to one or more embodiments of this specification are explained.

[0025] An entity association field set includes a plurality of entity association fields. Data holding entities of the plurality of entity association fields are the same and have different names.

[0026] A data holding entity is an entity that enjoys permission to data. For example, the data

holding entity can be a company or a country to which data belongs.

[0027] Data lineage is a relationship and dependency relationship between data elements. In a database, the data lineage can include table lineage, field lineage, and task lineage. A transfer path of a data element from a source to a target can be tracked based on the data lineage, to better understand a relationship between data and improve efficiency and accuracy of data management and application.

[0028] An entity model tree is a tree structure in which an entity is used as a node, and is used to describe a relationship between entities and a hierarchical structure. In computer science, the entity model tree is usually used to represent an entity relationship in a database. Each entity is a node of the tree, and a relationship between entities is an edge of the tree.

[0029] Most multi-national enterprises, in consideration of both development costs and operation and maintenance costs, will choose to store service data from multiple entities (e.g. multiple countries) in one data station. However, data peeling, data management and control, or data filing needs to be able to be quickly performed on data of different data holding entities, to respond to complex geopolitical changes and regional regulatory compliance challenges. Therefore, how to automatically identify data holding entities of hybrid multi-entity data becomes an urgent technical problem to be resolved.

[0030] Based on the foregoing content, embodiments of this specification provide a data processing method and device. An entity association field set of a seed dimension table is preset, so that a plurality of entity association fields of a service entity of the seed dimension table can be obtained. In addition, labeling information corresponding to a to-be-labeled table is determined based on an entity association field that is of the seed dimension table and that corresponds to a table field of the to-be-labeled table, so that a field that is of the to-be-labeled table and that is associated with the service entity of the seed dimension table can be accurately identified, to perform accurate data holding entity identification on the to-be-labeled table. In addition, data of each row in the to-be-labeled table is labeled based on labeling information that includes data holding entity information and that is recorded in the seed dimension table, to automatically perform data holding entity labeling on data of each row in the to-be-labeled table, so that entity identification can be performed on hybrid multi-entity data.

[0031] The following describes in detail the technical solutions of the embodiments of this specification with reference to the accompanying drawings.

[0032] FIG. 1 is a schematic diagram of an implementation environment of a data processing method according to an embodiment of this specification.

[0033] As shown in FIG. 1, the implementation environment 100 can include a terminal 110, a server 130, and a database 140.

5 **[0034]** The terminal 110 is connected to the server 130 through a wireless network or a wired network 120. The terminal 110 can be a tablet, a notebook computer, a desktop computer, etc., but is not limited thereto.

[0035] The terminal 110, that is, a background management end can store data or instructions for performing the data processing method described in this specification. The terminal 110 can
10 include a hardware device with a data information processing capability and a necessary program required for driving the hardware device to work. A background manager can work by using the terminal 110. For example, the background manager can configure a candidate field set, a derived dimension table, etc. on the terminal 110.

[0036] The server 130 is an independent physical server; or a server cluster or a distributed
15 system including a plurality of physical servers; or a cloud server that provides a cloud service, a cloud database, cloud computing, a cloud function, cloud storage, a network service, cloud communication, a middleware service, a domain name service, a security service, a content delivery network (CDN), big data, an artificial intelligence platform; etc. The server 130 provides a background service for an application running on the terminal 110.

20 **[0037]** The server 130 is provided with an integrated development platform. The integrated development platform is also referred to as an integrated development environment (IDE). The integrated development platform is an application used to provide a program development environment, and usually includes tools such as a code editor, a compiler, a debugger, and a graphical user interface. A developer can write program code on the integrated development
25 platform (that is, perform program development). An integrated development platform server can be a computing device specifically used by the integrated development platform to implement the data processing method. The server 130 can separately perform data communication with the terminal 110 and the database 140.

[0038] In addition, the server 130 can store the data or instructions for performing the data
30 processing method described in this specification. The server 130 can include a hardware device with a data information processing capability and a necessary program required for driving the

hardware device to work. Certainly, the server 130 can be only a hardware device with a data processing capability, or can be only a program that runs in the hardware device. In some embodiments, the server 130 can also serve as a plug-in and be deployed on the terminal 110. In this case, the server 130 stores data or instructions for performing a data processing method
5 corresponding to the terminal 110 described in this specification.

[0039] The database 140 can store data and/or instructions. In some embodiments, the database 140 can store a plurality of seed dimension tables and a to-be-labeled table. In some embodiments, the database 140 can store data and/or instructions used by the server 130 to perform the data processing method described in this specification. The terminal 110 and the server 130 may have
10 permission to access the database 140, and the terminal 110 and the server 130 can access, through a network, the data or instructions stored in the database 140. In some embodiments, the database 140 can be directly connected to the terminal 110 and the server 130. In some embodiments, the database 140 can be part of the server 130. In some embodiments, the database 140 can include a mass memory, a removable memory, a volatile read-write memory, a read-only memory (ROM)
15 or similar content, or any combination thereof. Example mass memories may include a non-transitory storage medium (non-transitory storage medium) such as a disk, an optical disk, or a solid-state disk. Example removable memories may include a flash drive, a floppy disk, an optical disc, a storage card, a zip disk, a magnetic tape, etc. A typical volatile read-write memory may include a random access memory (RAM). Example RAMs may include a dynamic RAM (DRAM),
20 a double data rate synchronous dynamic RAM (DDRSDRAM), a static RAM (SRAM), a thyristor RAM (T-RAM), a zero-capacitor RAM (Z-RAM), etc. Example ROMs may include a mask ROM (MROM), a programmable ROM (PROM), an erasable programmable ROM (PEROM), an electrically erasable programmable ROM (EEPROM), a compact disc (CD-ROM), a digital versatile disk ROM, etc.

[0040] A person skilled in the art can learn that there can be more or fewer terminals. For example, there is only one terminal, or there are dozens or hundreds of terminals, or there are more terminals. In this case, the implementation environment further includes another terminal. A quantity of terminals and a device type are not limited in this embodiment of this specification.

[0041] After the implementation environment in this embodiment of this specification is
30 described, the following describes an application scenario of this embodiment of this specification with reference to the implementation environment. In the following description process, the

terminal is also the terminal 110 in the implementation environment, and the server is also the server 130 in the implementation environment. The technical solutions provided in the embodiments of this specification can be applied to a data holding entity identification scenario of cross-border e-commerce application data or cross-border social application data, for example, an entity identification scenario of multi-country data of a cross-border e-commerce application.

[0042] For example, the technical solutions provided in this embodiment of this specification are applied to the entity identification scenario of the multi-country data of the cross-border e-commerce application. It is assumed that a to-be-labeled table is a historical transaction data table, data holding entity information, for example, country information of a plurality of pieces of row data corresponding to an entity association field is recorded in a plurality of seed dimension tables, an entity association field set of the seed dimension table includes a plurality of entity association fields, and the entity association field is used to associate an entity of the to-be-labeled table with an entity of the seed dimension table. The terminal 110 determines at least one table field of the to-be-labeled table. If an entity association field corresponding to the table field exists in an entity association field set of at least one seed dimension table, data holding entity information corresponding to data of each row in the to-be-labeled table is determined based on data holding entity information corresponding to an entity association field of at least one seed dimension table, and data of each row in the to-be-labeled table is labeled based on the determined data holding entity information.

[0043] It should be noted that the foregoing provides descriptions by using an example in which the technical solutions provided in this embodiment of this specification are applied to the entity identification scenario of the multi-country data of the cross-border e-commerce application. The technical solutions provided in this embodiment of this specification can also be applied to another proper entity identification scenario of multi-entity data. An implementation process and the above-mentioned description belong to the same inventive concept. Details are not described herein again.

[0044] It should be noted that steps in the data processing method in the example embodiments of this specification can be partially performed by a client and partially performed by a server, or can be all performed by a server or all performed by a client. This is not specifically limited in this specification.

[0045] Based on the implementation environment shown in FIG. 1, the following describes in

detail a data processing method and a data processing device provided in the embodiments of this specification with reference to FIG. 2 to FIG. 9. It should be noted that the implementation environment is merely shown for ease of understanding of the spirit and principle of this specification, and the embodiments of this specification are not limited in this aspect. On the contrary, the embodiments of this specification can be applied to any applicable scenario.

[0046] FIG. 2 is a schematic structural diagram of a data processing device 200 according to some embodiments of this specification. The data processing device 200 can perform a data processing method described in this specification. The data processing method is described in another part of this specification. The data processing device 200 can be a general-purpose computer or a dedicated computer. For example, the data processing device 200 can be a server, a personal computer, or a portable computer (for example, a notebook computer or a tablet), or can be another data processing device with a computing capability. Certainly, the data processing device can be the terminal 110 or the server 130 in FIG. 1, or can be a terminal device that is used by a plurality of developers to perform program development on an integrated development platform.

[0047] The data processing device in this specification can include one or more of the following components: a processor 210, a memory 220, an input apparatus 230, an output apparatus 240, and a bus 250. The processor 210, the memory 220, the input apparatus 230, and the output apparatus 240 can be connected through the bus 250.

[0048] The processor 210 can include one or more processing cores. The processor 210 is connected to all parts of the entire data processing device through various interfaces and lines, and performs the data processing method described in this specification by running or executing instructions, a program, a code set, or an instruction set stored in the memory 220 and invoking data stored in the memory 220. Optionally, the processor 210 can be implemented in at least one hardware form in a digital signal processor (DSP), a field-programmable gate array (FPGA), and a programmable logic array (PLA). The processor 210 can integrate one or a combination of a central processing unit (CPU), a graphics processing unit (graphics processing unit, GPU), a modem, etc. The CPU mainly processes an operating system, a user interface, an application, etc. The GPU is configured to be responsible for rendering and drawing display content. The modem is configured to process wireless communication. It can be understood that the modem may not be integrated into the processor 210, and is separately implemented by using a communication chip.

[0049] The memory 220 can include a random access memory (RAM), or can include a read-only memory (ROM). Optionally, the memory 220 includes a non-transitory computer-readable storage medium. The memory 220 can be configured to store instructions, a program, code, a code set, or an instruction set. The memory 220 can include a program storage area and a data storage area. The program storage area can store an instruction for implementing an operating system, an instruction for implementing at least one function (for example, a touch function, a sound playing function, or an image playing function), an instruction for implementing the following method embodiments, etc. The operating system can be an Android system, including a system deeply developed based on the Android system; or an IOS system, including a system deeply developed based on the IOS system; or another system.

[0050] To enable the operating system to distinguish between specific application scenarios of a third-party application, data communication between the third-party application and the operating system needs to be enabled, so that the operating system can obtain current scenario information of the third-party application at any time, and then perform targeted system resource adaptation based on a current scenario.

[0051] The input apparatus 230 is configured to receive input instructions or data, and the input apparatus 230 includes but is not limited to a keyboard, a mouse, a camera, a microphone, or a touch device. The output apparatus 240 is configured to output instructions or data, and the output apparatus 240 includes but is not limited to a display device, a speaker, etc. In an example, the input apparatus 230 and the output apparatus 240 can be disposed together, and the input apparatus 230 and the output apparatus 240 are touchscreens.

[0052] In addition, a person skilled in the art can understand that a structure of the data processing device shown in the accompanying drawings does not constitute a limitation on the data processing device, and the data processing device can include more or fewer components than those shown in the figure, or combine some components, or have different component arrangements. For example, the data processing device further includes components such as a radio frequency circuit, an input unit, a sensor, an audio circuit, a wireless fidelity (Wi-Fi) module, a power supply, and a Bluetooth module. Details are not described herein again.

[0053] FIG. 3 is a method flowchart of a data processing method according to some embodiments of this specification. The following describes the technical solutions of this specification with reference to FIG. 3. The technical solutions can be implemented by at least one

of the terminal 110 and the server 130 in FIG. 2. Specifically, the terminal 110 and/or the server 130 can have the structure described in FIG. 2. That is, the terminal 110 and/or the server 130 can be a data processing device for data processing, including at least one storage medium and at least one processor. The at least one storage medium includes at least one instruction set, and is
5 configured to perform the data processing method. The at least one processor is communicatively connected to the at least one storage medium. When the data processing device runs, the at least one processor can read the at least one instruction set, and perform the method in FIG. 3 based on an indication of the at least one instruction set. For a purpose of description only, an example in which the terminal 110 performs the data processing method is used for description in this
10 specification.

[0054] As shown in FIG. 3, in step S310, at least one table field of a to-be-labeled table and a plurality of seed dimension tables are determined.

[0055] In an example embodiment, the to-be-labeled table is a data table for a target service, for example, a data table for a target service in an offline data warehouse, and the plurality of seed
15 dimension tables are dimension tables of a service entity corresponding to the target service. The seed dimension table records a data holding entity to which the service entity belongs, service information, region information, etc. The target service can be a payment service, acquiring business, etc., or can be another proper service, for example, a membership service. The plurality of seed dimension tables corresponding to the target service are obtained from a database based
20 on identification information of the target service. For example, the plurality of seed dimension tables can include dimension tables such as a membership seed dimension table and a merchant dimension table.

[0056] Further, in an example embodiment, the terminal 110, that is, a background management end traverses all fields of the to-be-labeled table, and uses a traversed current field
25 of the to-be-labeled table as a table field. For example, it is assumed that the to-be-labeled table is a membership information table, the membership information table includes a plurality of fields, and the plurality of fields include fields such as a membership identifier field, a membership name field, and a membership card field. The terminal 110 traverses a plurality of fields of the to-be-labeled table. If the traversed current field is a membership identifier field, the membership
30 identifier field is used as a table field; or if the traversed current field is a membership name field, the membership name field is used as a next table field.

[0057] In step S320, if an entity association field corresponding to the table field exists in an entity association field set of at least one seed dimension table, labeling information corresponding to data of each row in the to-be-labeled table is determined from the at least one seed dimension table based on the corresponding entity association field.

5 **[0058]** In an example embodiment, the entity association field set includes a plurality of entity association fields, and the entity association field is used to associate an entity of the to-be-labeled table with an entity of the seed dimension table. For example, service entities corresponding to the plurality of entity association fields are the same and have different names. The terminal 110 compares the entity association field in the entity association field set of the seed dimension table with the table field. If a comparison result is that the entity association field and the table field are consistent, it indicates that the entity association field corresponding to the table field exists in the entity association field set of the seed dimension table; or if a comparison result is that the entity association field and the table field are inconsistent, it indicates that the entity association field corresponding to the table field does not exist in the entity association field set of the seed dimension table. For example, the plurality of seed dimension tables are traversed, to obtain an entity association field set of a traversed current seed dimension table, and an entity association field in the entity association field set is compared with the table field. If a comparison result is that the entity association field and the table field are consistent, it indicates that the entity association field corresponding to the table field exists in the entity association field set.

20 **[0059]** For example, it is assumed that the to-be-labeled table is a membership information table, the membership information table includes a membership identifier field, a membership name field, and a membership card field, and an entity association field set of the membership seed dimension table includes a membership name field. An entity association field set of a traversed membership seed dimension table is obtained, and a membership entity association field in the entity association field set is compared with the membership name field in the membership information table. If a comparison result is that the membership entity association field and the membership name field are consistent, it indicates that an entity association field corresponding to the membership name field exists in the entity association field set.

30 **[0060]** Further, in an example embodiment, the seed dimension table records labeling information of a plurality of pieces of row data corresponding to the entity association field, the labeling information includes information on data holding entity, the entity association field set of

the seed dimension table includes a plurality of entity association fields, and service entities corresponding to the plurality of entity association fields are the same and have different names. Because the plurality of entity association fields correspond to the same service entity, and the same service entity belongs to the same data holding entity, data entities corresponding to the plurality of entity association fields are the same and have different names. The terminal 110 determines, from the at least one seed dimension table based on the entity association field corresponding to the table field in the entity association field set of the at least one seed dimension table, the labeling information, for example, data holding entity information corresponding to data of each row in the to-be-labeled table. For example, each piece of row field data corresponding to the entity association field of the seed dimension table is matched with each piece of row field data of the table field of the to-be-labeled table. If one piece of row field data of the table field is matched, it indicates that labeling information of corresponding row data of the row field data is labeling information of row field data of a corresponding entity association field.

[0061] For example, it is assumed that the to-be-labeled table is a membership information table, the membership information table includes a membership identifier field, a membership name field, and a membership card field, and an entity association field set of the membership seed dimension table includes a membership name field. Each piece of row field data corresponding to the membership name field of the membership seed dimension table is matched with each piece of row field data of the membership name field of the membership information table, and labeling information corresponding to data of each row of the membership information table is determined based on a matching result. That is, labeling information corresponding to matched row field data of the membership seed dimension table is used as labeling information of corresponding row data of the membership information table.

[0062] In step S330, data of each row in the to-be-labeled table is labeled based on the labeling information.

[0063] In an example embodiment, the labeling information includes information on data holding entity. The terminal 110 performs data holding entity labeling on data of each row in the to-be-labeled table based on the determined data holding entity information corresponding to data of each row in the to-be-labeled table. For example, the data holding entity information corresponding to data of each row in the to-be-labeled table is a data holding entity label.

[0064] For example, it is assumed that the to-be-labeled table is a membership information

table, the membership information table includes a membership name field, and an entity association field set of the membership seed dimension table includes a membership name field. The terminal 110 labels data of each row of the membership information table based on determined data holding entity information of row field data that corresponds to data of each row of the membership information table and that is in the membership seed dimension table.

[0065] According to the technical solutions in the example embodiment in FIG. 3, the entity association field set of the seed dimension table is preset, so that a plurality of entity association fields of a service entity of the seed dimension table can be obtained. In addition, the labeling information corresponding to the to-be-labeled table is determined based on the entity association field that is of the seed dimension table and that corresponds to the table field of the to-be-labeled table, so that a field that is of the to-be-labeled table and that is associated with the service entity of the seed dimension table can be accurately identified, to perform accurate data holding entity identification on the to-be-labeled table. In addition, data of each row in the to-be-labeled table is labeled based on the labeling information that includes the data holding entity information and that is recorded in the seed dimension table, to automatically perform data holding entity labeling on data of each row in the to-be-labeled table, so that entity identification can be performed on hybrid multi-entity data.

[0066] In addition, in some embodiments, the data processing method further includes: determining an upstream table of the to-be-labeled table based on a table lineage relationship if no entity association field corresponding to the table field of the to-be-labeled table exists in an entity association field set of each of the plurality of seed dimension tables; and performing table labeling on the to-be-labeled table based on labeling information of the upstream table.

[0067] For example, if no entity association field corresponding to the table field exists in the entity association field set of each of the plurality of seed dimension tables, it indicates that the to-be-labeled table is a data table that cannot be directly labeled, the upstream table of the to-be-labeled table is determined based on the table lineage relationship, labeling information of row data that corresponds to data of each row in the to-be-labeled table and that is in the upstream table is determined based on a field lineage propagation link between the upstream table and a field of the to-be-labeled table, and the to-be-labeled table is labeled based on the labeling information corresponding to the row data in the upstream table.

[0068] According to the technical solution in the example embodiments, the upstream table of

the to-be-labeled table is determined based on the table lineage relationship; and table labeling is performed on the to-be-labeled table based on the labeling information of the upstream table, to label a data table that cannot be labeled, thereby improving a coverage rate of labeling the data table.

5 **[0069]** Further, in some example embodiments, the data processing method further includes: if no entity association field corresponding to the table field of the to-be-labeled table exists in an entity association field set of each of the plurality of seed dimension tables, determining whether a derived field in a derived dimension table includes the table field; and if the derived field includes the table field, determining labeling information corresponding to data of each row of the table
10 field based on an association relationship between the derived dimension table and the seed dimension table. The derived dimension table is a data table derived by associating the seed dimension table with another data table. For example, a membership card derived dimension table is a data table that is related to an entity and a service and that is derived by associating the seed dimension table with another data table. That is, when the entity association field in the to-be-labeled table cannot be directly associated with the seed dimension table, the to-be-labeled table
15 and the seed dimension table can be indirectly associated by using another data table associated with the to-be-labeled table, and a data holding entity and a service of the to-be-labeled table can be labeled.

[0070] For example, the data holding entity and the service of the to-be-labeled table can be
20 identified based on an association relationship between the to-be-labeled table and the derived dimension table. For example, it is assumed that the derived dimension table is a data table derived by associating the membership seed dimension table and a card data table, the to-be-labeled table is a membership card data table, and a card number field in a membership card cannot be directly associated with a membership field in the membership seed dimension table. Labeling information
25 of a membership field in a membership seed table corresponding to data of each row in the membership card data table is determined based on an association relationship between a membership field in the membership seed dimension table and a card number field in the card data table in the derived dimension table. That is, labeling information of data of each row in the membership card data table is determined based on the labeling information of the membership
30 field in the corresponding membership seed table.

[0071] According to the technical solutions in the example embodiments, when the to-be-

labeled table cannot be directly associated with the seed dimension table, the to-be-labeled table and the seed dimension table are indirectly associated by using the derived dimension table, and data of each row in the to-be-labeled table is labeled.

[0072] In addition, in some example embodiments, after data of each row in the to-be-labeled table is labeled, the data processing method further includes: performing aggregation processing on the labeling information of data of each row in the to-be-labeled table. For example, it is assumed that the to-be-labeled table includes 100 rows of data. Labeling information of the 100 rows of row data is aggregated, to obtain labeling information of the entire to-be-labeled table.

[0073] According to the technical solutions in the example embodiments, aggregation processing is performed on the labeling information of data of each row in the to-be-labeled table, to obtain the labeling information of the entire to-be-labeled table.

[0074] FIG. 4 is a schematic flowchart of determining labeling information of a to-be-labeled table according to some embodiments of this specification.

[0075] As shown in FIG. 4, in step S410, a connection query is performed on data of each row of an entity association field of a target seed dimension table in at least one seed dimension table and data of each row of a table field of a to-be-labeled table.

[0076] In an example embodiment, a connection JOIN query is to jointly complete a query operation on two or more tables, two tables on which a connection query is performed have mutually associated fields, and a query operation is performed on the two tables by using the mutually associated fields of the two tables. The target seed dimension table is determined from the at least one seed dimension table, and the connection query is performed on data of each row of the entity association field of the target seed dimension table and data of each row of the table field of the to-be-labeled table.

[0077] For example, it is assumed that the to-be-labeled table is a membership information table, the membership information table includes a membership identifier field, a membership name field, and a membership card field, and an entity association field set of the membership seed dimension table includes a membership name field. A connection query is performed on each piece of row field data corresponding to the membership name field of the membership seed dimension table and each piece of row field data of the membership name field of the membership information table.

[0078] In step S420, the labeling information corresponding to data of each row in the to-be-

labeled table is determined based on a result of the connection query.

[0079] In an example embodiment, the labeling information corresponding to data of each row in the to-be-labeled table is determined based on a query result of the connection query performed on data of each row in the entity association field of the target seed dimension table and data of each row in the table field of the to-be-labeled table.

[0080] For example, it is assumed that the to-be-labeled table is a membership information table, a table field of the membership information table is a membership name field, an entity association field set of a membership seed dimension table includes a membership name field, and the labeling information includes data holding entity information. Data holding entity information corresponding to data of each row in the to-be-labeled table is determined based on a query result of a connection query performed on data of each row of the membership name field of the membership seed dimension table and data of each row of the membership name field of the membership information table. That is, data holding entity information that corresponds to row field data of the membership seed dimension table and that is obtained through a connection query is used as data holding entity information of corresponding row data in the membership information table.

[0081] According to the technical solutions in the example embodiment in FIG. 4, the connection query is performed on data of each row of the entity association field of the target seed dimension table and data of each row of the table field of the to-be-labeled table, and the labeling information corresponding to data of each row in the to-be-labeled table is determined based on the result of the connection query, to accurately obtain the data holding entity information of data of each row in the to-be-labeled table, thereby identifying a data holding entity of data of each row in the to-be-labeled table.

[0082] In addition, in some embodiments, each seed dimension table is ranked based on a priority of each of the at least one seed dimension table; and the target seed dimension table is determined based on a ranking order of the at least one seed dimension table. For example, the priority of each seed dimension table is preset, each seed dimension table is ranked based on a priority of each seed dimension table in a seed dimension table group corresponding to a target service, and a seed dimension table ranked the first is selected as the target seed dimension table.

[0083] According to the technical solutions in the example embodiment, the target seed dimension table is determined based on a ranking order of the priority of the seed dimension table,

and a seed dimension table with a highest priority can be selected and labeled, to improve data processing efficiency of row labeling.

[0084] In addition, in some example embodiments, the data processing method further includes: determining an entity model tree of a plurality of data tables in a target database; and
5 generating, based on the entity model tree, a seed dimension table corresponding to the plurality of data tables in the target database. The entity model tree is a tree structure in which an entity is used as a node, and is used to describe a relationship between entities and a hierarchical structure. The entity model tree is usually used to represent an entity relationship in a database. Each entity is a node of the tree, and a relationship between entities is an edge of the tree. The entity model
10 tree can help better understand and manage a data model and more effectively implement data access and management.

[0085] The node of the entity model tree is a basic service entity in a data warehouse. Various data tables in the data warehouse are events, configurations, and statistics information related to these service entities. The edge of the entity model tree is a basis for determining a data holding
15 entity for a service entity. For example, each merchant has one settled entity. It can be considered that data of a table in which a merchant ID is a primary key is from/belongs to the settled entity. Acquiring and settlement are related to the merchant. Therefore, a source/belonging entity of acquiring data and settlement data need to be related to the settled entity of the merchant. As shown in FIG. 9, the entity model tree includes a merchant entity, a membership entity, etc., the merchant
20 entity corresponds to a settlement table and an acquiring table, the membership entity corresponds to a membership card asset table, an institution flow table, an accounting flow table, etc., and the merchant entity and the membership entity correspond to a fund account table, a fund product table, and a payment document.

[0086] Further, the seed dimension table corresponding to the plurality of data tables in the
25 target database is generated based on the entity model tree. The seed dimension table records a entity to which the data belongs, a service, and region information. The plurality of seed dimension tables can form a dimension table group. The dimension table group is a summary of service entities in the same service domain. In an implementation process, a priority is specified for a plurality of dimension tables in a dimension table group, so that the selected seed dimension table
30 more matches a definition of a service. For example, priorities of a merchant dimension table group are as follows: a priority of a merchant dimension table > a priority of an acquiring dimension

table; priorities of a user group are as follows: a priority of a membership dimension table > a priority of remittance > a priority of a membership card asset dimension table; priorities of a fund account group are as follows: a priority of an account dimension table > a priority of transit funds > a priority of a settlement dimension table > a priority of a payment dimension table > a priority of a fund product dimension table; priorities of a common group are as follows: a priority of a fund accounting entity dimension table & a priority of a marketing financial entity dimension table. The common group indicates that an existing field in the group of dimension tables can indicate a entity to which data belongs, and labeling can be directly used by using a entity label in the table.

[0087] According to the example embodiments, an entity model tree of the database is constructed, to refine a core entity model of the database, and simplify a entity identification scenario.

[0088] Further, in an example embodiment, for covered row data that is in the to-be-labeled table and that is covered by the result of the connection query, labeling information of the corresponding covered row data in the to-be-labeled table is determined based on labeling information of data of each row of the entity association field; and for uncovered row data that is in the to-be-labeled table and that is not covered by the result of the connection query, the uncovered row data is labeled based on a non-standard type of the uncovered row data. The non-standard type includes a non-standard format and non-standard content.

[0089] For example, it is assumed that the to-be-labeled table includes 100 rows of data. If the result of the connection query covers 70 rows of data in the to-be-labeled table, for the 70 rows of covered row data, labeling information of row data that is of the entity association field of the seed dimension table and that corresponds to the covered row data is used as labeling information of the covered row data. For 30 rows of uncovered row data that is not covered by the result of the connection query, a non-standard type of the uncovered row data is determined, and the uncovered row data is labeled based on the non-standard type of the uncovered row data.

[0090] Further, if a data format of the uncovered row data is not a standard format, it is determined that the non-standard type of the uncovered row data is a non-standard format; or if a data format of the uncovered row data is a standard format, and data content of the uncovered row data is invalid content, it is determined that the non-standard type of the uncovered row data is non-standard content. The non-standard format and the non-standard content are labeled, to resolve a problem that a coverage rate is low. When the to-be-labeled table includes a large amount of row

data that cannot be labeled, a problem may occur in the table. For example, the row data is not tested. Therefore, when there is no entity association field, whether a format of data is standard is checked for the uncovered row data. If the format is not standard, a row label of a non-standard format is labeled. In addition, when the to-be-labeled table includes a large quantity of rows that cannot be labeled, although a data format of some uncovered row data is standard, data content of the uncovered row data is not necessarily valid. Therefore, uncovered row data with invalid content needs to be labeled with a row label of non-standard content.

[0091] According to the technical solutions in the example embodiment, the uncovered row data is labeled based on the non-standard type of the uncovered row data, for example, a non-standard format or non-standard content, to improve a row coverage rate of labeling the to-be-labeled table.

[0092] In addition, in an example embodiment, before the labeling information corresponding to data of each row in the to-be-labeled table is determined, the data processing method further includes: determining an entity association field set of each seed dimension table. The entity association field set includes a candidate field subset and a lineage field subset. Service entities corresponding to a plurality of entity association fields in the entity association field set are the same but have different names. Therefore, data holding entities corresponding to the plurality of entity association fields in the entity association field set are the same, the seed dimension table is associated with a field in the to-be-labeled table by using the entity association field, and corresponding row data in the to-be-labeled table is labeled based on a data holding entity corresponding to a data row of the entity association field. However, the same service entity in an offline table usually has different names, a candidate field subset of the same service entity of different names needs to be established, and a plurality of candidate fields in the candidate field subset are fields of the same service entity that have different names. For example, for a user_id field, fields having different names such as login_profile_id, payee_id, usr_id, userid, and account_no exist in different tables. The fields of the same service entity that have different names can be preconfigured in a visual graphical interface, to form a candidate field subset corresponding to the service entity, so that the seed dimension table can be associated with more data tables.

[0093] According to the example embodiments, a candidate field set is set for a core entity association field that appears at a high frequency, to improve efficiency and accuracy of labeling the data table.

[0094] Further, because a name of the same service entity usually changes in different data tables, aliases of all fields cannot be exhaustively listed by presetting a candidate set. In addition, because some unclear names may correspond to different service entities, even if a field of the name is associated, a seed dimension table cannot be associated. For example, for the user_id field, it referred to as account_no in a cross-border e-commerce settlement service. However, the account_no field can also correspond to a collection account entity, or can correspond to a fund account entity, an institution account entity, etc. Therefore, label propagation can be performed based on lineage, to obtain a lineage field subset of a plurality of fields of the same service entity. The following describes in detail determining of a lineage field with reference to FIG. 5.

[0095] FIG. 5 is a schematic flowchart of determining a lineage field according to some embodiments of this specification.

[0096] As shown in FIG. 5, in step S510, a lineage graph of a plurality of data tables is constructed based on data lineage of each of the plurality of data tables.

[0097] In an example embodiment, a lineage graph, for example, a directed acyclic graph (DAG) lineage graph of a plurality of data tables in a database is constructed based on data lineage of the plurality of data tables. For example, a structured query language (SQL) statement corresponding to the plurality of data tables in the database is parsed, to obtain field-level and table-level lineage relationships of the plurality of data tables. The field-level and table-level lineage relationships of the plurality of data tables are then converted into a lineage graph, for example, a DAG lineage graph.

[0098] In step S520, an associated data table associated with a seed dimension table in the plurality of data tables is determined. The associated data table includes a reference lineage field corresponding to an entity association field of the seed dimension table.

[0099] In an example embodiment, the associated data table associated with the seed dimension table in the plurality of data tables is determined based on a lineage relationship between the entity association field in the seed dimension table and the field of the data table. The associated data table includes the reference lineage field corresponding to the entity association field of the seed dimension table. Further, an association relationship of the reference lineage field in the associated data table can be manually determined, to further improve accuracy of the determined lineage field.

[0100] In step S530, a lineage field corresponding to the reference lineage field in a

downstream table of each associated data table is determined based on a reference lineage field in each associated data table and the lineage graph.

5 **[0101]** In an example embodiment, the downstream table of each associated data table is determined from the plurality of data tables based on the lineage graph of the plurality of data tables in the database, and the lineage field corresponding to the reference lineage field in the downstream table is determined based on the data lineage relationship and is continuously propagated downstream, so that a lineage field subset of each seed dimension table can be obtained based on an obtained reference lineage field in the upstream table and the obtained lineage field of the downstream table.

10 **[0102]** In some other example embodiments, an initial lineage field of each data table is determined based on a candidate field set of each of the plurality of data tables; the downstream table of the associated data table is determined from the plurality of data tables based on a table lineage relationship; and if the downstream table inherits the reference lineage field in the associated data table, an association relationship of the initial lineage field of the downstream table is corrected based on the reference lineage field. One exemplary implementation procedure of the above-mentioned steps comprises the following steps:

[0103] (1) Perform an initial round of automatic labeling on the plurality of data tables in the database based on a candidate field subset, and determine the initial lineage field of each data table.

[0104] (2) Use, as the associated data table, a data table whose association relationship is determined to be right; and determine the downstream table of the associated data table based on the table lineage relationship. For example, the associated data table is determined as a downstream table of a primary table (a drive table) based on a table-to-table lineage relationship, and the associated data table includes the reference lineage field corresponding to the entity association field of the seed dimension table.

25 **[0105]** (3) Determine, based on a column lineage relationship, whether the downstream table inherits an entity association field of the upstream table; and if the downstream table inherits the entity association field of the upstream table, correct the association relationship of the initial lineage field of the downstream table based on the reference lineage field. For example, an inheritance manner in which the downstream table inherits the reference lineage field in the associated data table is determined; and the initial lineage field of the downstream table is corrected based on the reference lineage field if the inheritance manner is a direct dependency relationship.

[0106] According to the technical solutions in the foregoing embodiment, the initial lineage field of the data table is corrected based on a lineage relationship, to more accurately determine the lineage field of the associated data table associated with the seed dimension table, thereby more accurately determining the lineage field subset of the seed dimension table.

5 [0107] In step S540, the lineage field subset of each seed dimension table is determined based on the reference lineage field and the lineage field.

[0108] In an example embodiment, the reference lineage field in the associated data table and the corresponding lineage field of the downstream table are summarized, to obtain the lineage field subset of the seed dimension table associated with the associated data table.

10 [0109] According to the technical solutions in the example embodiment in FIG. 5, the lineage field of the associated data table associated with the seed dimension table is determined based on the lineage relationship of the plurality of data tables in the database, and the lineage field of the entity association field of each seed dimension table is determined from the downstream table, to obtain the lineage field corresponding to the entity association field of the seed dimension table in

15 the database.

[0110] FIG. 6 is a diagram of a functional architecture of a data processing device according to some embodiments of this specification.

[0111] As shown in FIG. 6, a functional module of the data processing device includes a label management module 610, a labeling management module 620, a scheduling engine 630, a labeling

20 task engine 640, and a rule engine 650. The label management module 610 is configured to manage labeling information, that is, a label of a seed dimension table, and the labeling management module 620 is configured to process a labeling task and a labeling result. The scheduling engine 630 is configured to process scheduling of the labeling task. The labeling task engine 640 is configured to process table labeling, row labeling, and field labeling. The rule engine 650 is

25 configured to select a proper rule from a dimension table rule, an expression rule, and a lineage rule for labeling processing.

[0112] The label management module 610 includes a label processing module and a label rule processing module. The label processing module includes a label definition unit and a label classification unit. The label definition unit is configured to define the labeling information of the

30 seed dimension table, that is, the label. The label classification unit is configured to classify types of the labeling information of the seed dimension table, for example, a data holding entity, service

information, and region information. The label rule processing module includes a dimension table rule processing unit, a structure rule processing unit, and a rule iteration unit. The dimension table rule processing unit is configured to perform configuration processing on a naming rule of the seed dimension table, for example, a naming rule of a table name and a field name. The structure rule processing unit is configured to process a rule of a table structure of the seed dimension table, for example, a sequence structure of fields of the table. The rule iteration unit is configured to perform iteration updating on the dimension table rule and a dimension table structure rule.

[0113] The labeling association module 620 includes a labeling task processing module and a labeling result processing module. The labeling task processing module includes a labeling item processing unit, a labeling procedure processing unit, a labeling task processing unit, and a labeling task execution unit. The labeling item processing unit is configured to manage a labeling item, for example, manage a labeling item of a target database. The labeling procedure processing unit is configured to manage a labeling procedure. The labeling task processing unit is configured to manage processing of the labeling task. The labeling task execution unit is configured to manage execution of the labeling task. The labeling result processing module includes a manual labeling unit and a labeling result review unit, the manual labeling unit is configured to manage manual labeling, and the labeling result review unit is configured to manage a review result of the labeling result.

[0114] The scheduling engine 630 includes a metadata loading unit, an instance creation unit, an instance execution unit, and a labeling result query unit. The metadata loading unit is configured to query rule metadata and a labeling task from the label management module, the rule metadata can include a dimension table rule and dimension table metadata of the seed dimension table, and the labeling task can include task data of a to-be-labeled table that currently needs to be labeled. The instance creation unit is configured to create a labeling task instance based on the data of the seed dimension table and the data of the to-be-labeled table that are obtained by the metadata loading unit. The instance execution unit is configured to execute the created labeling task instance, to generate the labeling result. The labeling result query unit is configured to query the generated labeling result.

[0115] The labeling task engine 640 includes a labeling engine routing module, a table labeling module, a row labeling module, and a field labeling module. The labeling engine routing module is configured to route the labeling task to the table labeling module, the row labeling module, or

the field labeling module based on types of different labeling tasks. For example, a table labeling task is routed to the table labeling module, and a row labeling task is routed to the row labeling module. The table labeling module is configured to label an entire table. The row labeling module is configured to label row data in the table. The field labeling module is configured to label a field in the table, for example, label a sensitive field. The table labeling module, the row labeling module, and the field labeling module each include an instance generation unit, an instance execution unit, and a result query unit. Each instance generation unit is configured to generate a corresponding table labeling task instance, a corresponding row labeling task instance, and a corresponding field labeling task instance. Each instance execution unit is configured to execute the corresponding table labeling task instance, the corresponding row labeling task instance, and the corresponding field labeling task instance. Each result query unit is configured to query a corresponding table labeling result, a corresponding row labeling result, and a corresponding field labeling result.

[0116] The rule engine 650 includes a rule routing module, a dimension table engine module, an expression engine module, and a lineage engine module. The rule routing module is configured to select different rules for different labeling tasks, to determine an entity association field corresponding to a table field of the to-be-labeled table. For example, different labeling instances are routed to the corresponding dimension table engine module, the corresponding expression engine module, and the corresponding lineage engine module, and the entity association field corresponding to the table field of the to-be-labeled table is searched for through corresponding table rule matching, expression matching, or lineage relationship matching. A rule parsing unit of the dimension table engine module is configured to: determine a seed dimension table corresponding to the to-be-labeled table, and perform table rule parsing on the seed dimension table, to obtain a field corresponding to the seed dimension table. The rule execution unit is configured to: match the field of the seed dimension table with the table field of the to-be-labeled table, to obtain the entity association field corresponding to the table field of the seed dimension table, and determine the labeling information of the to-be-labeled table based on the entity association field. A rule parsing unit of the expression engine module is configured to parse an expression rule corresponding to the labeling task. The rule execution unit matches the table field of the to-be-labeled table with a candidate field of the seed dimension table based on a regular expression obtained through parsing, to obtain the entity association field corresponding to the table field of the seed dimension table, and determines the labeling information of the to-be-labeled

table based on the entity association field. A rule parsing unit of the lineage engine module is configured to parse a lineage relationship rule corresponding to the labeling task. The rule execution unit is configured to: match the table field of the to-be-labeled table with a lineage field of the seed dimension table based on a lineage relationship obtained through parsing, to obtain the entity association field corresponding to the table field of the seed dimension table, and determine the labeling information of the to-be-labeled table based on the entity association field.

[0117] According to the technical solutions in the example embodiment in FIG. 6, the labeling engine routing module is set, to label both a data table and row data in the data table. In addition, the rule routing module is set, to determine the entity association field corresponding to the table field of the to-be-labeled table in the seed dimension table in a plurality of manners, so that the labeling information of the to-be-labeled table can be determined in a plurality of manners, and a coverage rate of data labeling can be improved.

[0118] FIG. 7 is a schematic flowchart of a data processing method according to some other embodiments of this specification.

[0119] As shown in FIG. 7, in step S710, labeling is performed based on a preset table rule.

[0120] In an example embodiment, the preset table rule is mainly used to exclude, based on a source item of a to-be-labeled table, a data table whose table permission cannot be applied for and for which data holding entities cannot be distinguished due to an item such as an external data item or a privacy protection item. A table that does not need to be labeled is identified through labeling based on the preset table rule, and further labeling processing is performed on only a data table that needs to be labeled.

[0121] In step S720, row labeling is performed based on an association relationship.

[0122] In an example embodiment, row data in the to-be-labeled table is labeled based on an association relationship between a table field in the to-be-labeled table and an entity association field in a seed dimension table. For example, a data holding entity of the row data is labeled. That is, the row data in the to-be-labeled table is labeled with correct labeling information such as a data holding entity and service information based on a determined entity association relationship between the table field of the to-be-labeled table and the entity association field of the seed dimension table.

[0123] In step S730, it is determined that a row coverage rate is 100%.

[0124] In an example embodiment, if a row coverage rate of the to-be-labeled table is 100%

in a specified associated column of the entity association field of the seed dimension table, or a row coverage rate of the to-be-labeled table is 100% in an identified associated column of a plurality of entity association fields of a plurality of seed dimension tables, a labeling result of the to-be-labeled table is directly determined. For example, the row coverage rate of the to-be-labeled table is 100% in a case of an identity card field of a membership dimension table, or the row coverage rate of the to-be-labeled table is 100% in a case of an identity card field of a membership dimension table and a card number field of the membership dimension table.

[0125] In step S740, there is an associated column, and a row coverage rate is less than 100%.

[0126] In an example embodiment, if there is an associated column between the to-be-labeled table and the seed dimension table, and the row coverage rate of the to-be-labeled table is less than 100, it is determined that there is uncovered row data in the to-be-labeled table. For example, it is assumed that the to-be-labeled table includes 100 rows of data. If a row labeling result covers 70 rows of data in the to-be-labeled table, and the row coverage rate is 70% and is less than 100%, for 30 rows of uncovered row data, a non-standard type of the uncovered row data is determined, and the uncovered row data is labeled based on the non-standard type of the uncovered row data.

[0127] In step S750, data in a non-standard format is labeled.

[0128] In an example embodiment, the non-standard format is labeled to resolve a problem that the coverage rate is low. When the to-be-labeled table includes a large amount of row data that cannot be labeled, a problem may occur in the table. For example, the row data is not tested. Therefore, when there is no entity association field, whether a format of data is standard is checked for the uncovered row data. If the format is not standard, a row label of a non-standard format is labeled. For example, if a seed dimension table includes a mobile phone number field, the mobile phone number field of the seed dimension table corresponds to an 11-digit pure digit character string that starts with 1, and whether uncovered row data corresponding to the mobile phone number field in the to-be-labeled table is in a non-labeled format is determined based on a preset regular rule corresponding to the seed dimension table.

[0129] In step S760, data with non-standard content is labeled.

[0130] In an example embodiment, when the to-be-labeled table includes a large quantity of rows that cannot be labeled, although a data format of some uncovered row data is standard, data content of the uncovered row data is not necessarily valid. Therefore, uncovered row data with invalid content needs to be labeled with a row label of non-standard content.

[0131] In step S770, no associated column is identified, and the row coverage rate is 0.

[0132] In an example embodiment, if no entity association field corresponding to the table field of the to-be-labeled table exists in an entity association field set of each of the plurality of seed dimension tables, it is determined that no associated column is identified, and the row coverage rate is 0.

[0133] In step S780, table labeling is performed.

[0134] In an example embodiment, for a data table that cannot be labeled, table labeling can be performed based on a table lineage relationship.

[0135] For example, data tables such as a summary table, an encryption masking table, an internal system table, an external data table, and a manual dimension table cannot be labeled. The summary table is used as an example. An upstream table of the summary table is determined based on the table lineage relationship, labeling information of corresponding row data in an upstream table corresponding to data of each row of the summary table is determined based on a field lineage propagation link between the upstream table and the summary table, and the summary table is labeled based on the labeling information of the corresponding row data in the upstream table.

[0136] In step S790, labeling is performed based on a post table rule.

[0137] In an example embodiment, a type of the to-be-labeled table is determined based on the post table rule. That is, the type of the to-be-labeled table on which row labeling cannot be performed is determined. For example, the type of the to-be-labeled table is determined based on information such as a table name and a lineage relationship. The type of the to-be-labeled table on which row labeling cannot be performed includes a summary table, an encryption masking table, an internal system table, an external data table, and a manual dimension table.

[0138] According to the technical solutions in the foregoing embodiment, data of each row in the to-be-labeled table is labeled based on the association relationship between the to-be-labeled table and the field of the seed dimension table, and data holding entity labeling can be automatically performed on data of each row in the to-be-labeled table, so that entity identification can be performed on hybrid multi-entity data. In addition, for a data table that cannot be labeled, rules such as a preset table rule, a post table rule, a non-standard format rule, and a non-standard content rule are set for labeling, so that a coverage rate of data labeling can be improved.

[0139] FIG. 8 is a schematic flowchart of a data processing method according to some other embodiments of this specification.

- [0140] As shown in FIG. 8, in step S805, all seed dimension tables are queried, and are denoted as a dimension table set DS.
- [0141] In an example embodiment, all seed dimension tables corresponding to a target database are queried, and are denoted as a dimension table set DS.
- 5 [0142] In step S810, all fields in a to-be-labeled table are queried, and are denoted as a field set CS.
- [0143] In step S815, the field set CS is traversed, and a traversed current field is denoted as C.
- [0144] In an example embodiment, the field set CS of the to-be-labeled table is traversed, and it is determined that the traversed current field is the table field C.
- 10 [0145] In step S820, the dimension table set DS is traversed, and a traversed current seed dimension table is denoted as D.
- [0146] In step S825, whether an entity association field set of D includes the table field C is determined.
- [0147] In an example embodiment, whether an entity association field set of the traversed current seed dimension table D includes the table field C is determined; step S830 is performed if
- 15 the table field C is included; or step S820 is performed if the table field C is not included.
- [0148] In step S830, a dimension table list associated with C is added.
- [0149] In an example embodiment, if the entity association field set of the current seed dimension table D includes the table field C, the current seed dimension table D is added to the
- 20 dimension table list associated with the table field C. The dimension table list is a set of a plurality of dimension tables. The dimension table list is set to ensure that a table can also be identified by using another entity association field when one entity association field is missing.
- [0150] In step S835, whether C is associated with a seed dimension table is determined.
- [0151] In an example embodiment, whether the table field C is associated with a seed
- 25 dimension table is determined, and step S840 is performed if the table field C is associated with a seed dimension table; or step S815 is performed if the table field C is associated with no seed dimension table.
- [0152] In step S840, seed dimension tables in the dimension table list associated with the table field C are ranked based on priorities.
- 30 [0153] In an example embodiment, the seed dimension tables in the dimension table list associated with the table field C are ranked based on preset priorities of the seed dimension tables.

[0154] In step S845, a seed dimension table with a highest priority is selected as the seed dimension table associated with the table field C.

[0155] According to the technical solutions in the foregoing example embodiment, a plurality of seed dimension tables associated with a table field of the to-be-labeled table are determined based on a relationship between an entity association field in an entity association field set of the seed dimension table and the table field of the to-be-labeled table, and the seed dimension table with the highest priority is selected for labeling, thereby improving data processing efficiency of row labeling.

[0156] In conclusion, according to the data processing method and the data processing device provided in this specification, labeling information corresponding to data of each row in the to-be-labeled table is determined based on an entity association field that is of the seed dimension table and that corresponds to the table field of the to-be-labeled table, so that a field that is of the to-be-labeled table and that is associated with the service entity of the seed dimension table can be accurately identified, to perform accurate data holding entity identification on the to-be-labeled table. In addition, data of each row in the to-be-labeled table is labeled based on labeling information that includes data holding entity information and that is recorded in the seed dimension table, to automatically perform data holding entity labeling on data of each row in the to-be-labeled table, so that entity identification can be performed on hybrid multi-entity data.

[0157] Another aspect of this specification provides a non-transitory storage medium. The non-transitory storage medium stores at least one group of executable instructions used to perform data processing. When the executable instructions are executed by a processor, the executable instructions instructs the processor to implement steps of the data processing method in this specification. In some possible implementations, aspects of this specification can further be implemented in a form of a program product, including program code. When the program product runs on a data processing device 200, the program code is used to enable the data processing device 200 to perform steps of the data processing method described in this specification. The program product used to implement the above-mentioned method can be a portable compact disk read-only memory (CD-ROM), including program code, and can run on the data processing device 200.

[0158] However, the program product in this specification is not limited thereto. In this specification, a readable storage medium can be any tangible medium that includes or stores a program, and the program can be used by or in combination with an instruction execution system.

The program product can be any combination of one or more readable media. The readable medium can be a readable signal medium or a readable storage medium. The readable storage medium can be, for example, an electrical system, apparatus, or device, a magnetic system, apparatus, or device, an optical system, apparatus, or device, an electromagnetic system, apparatus, or device, an infrared system, apparatus, or device, or a semiconductor system, apparatus, or device, or any combination thereof. More specific examples of the readable storage medium include: an electrical connection including one or more conducting wires, a portable disk, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage component, a magnetic storage component, or any proper combination thereof. The computer-readable storage medium can include a data signal propagated in a baseband or as part of a carrier, and carries readable program code. The propagated data signal can be in a plurality of forms, and includes but is not limited to an electromagnetic signal, an optical signal, or any proper combination thereof. The readable storage medium can further be any readable medium other than the readable storage medium, and the readable medium can be used to send, propagate, or transmit a program used by or in combination with an instruction execution system, apparatus, or component. The program code included in the readable storage medium can be transmitted in any proper medium, including but not limited to a wireless medium, a wired medium, optical cable, an RF medium, or any proper combination thereof. Program code for performing an operation of this specification can be written in any combination of one or more program design languages. The program design language includes an object-oriented program design language, for example, Java or C++, and further includes a conventional procedural program design language, for example, "C" language, or a similar program design language. The program code can be completely executed on the data processing device 200, partially executed on the data processing device 200, executed as an independent software package, partially executed on the data processing device 200 and partially executed on a remote computing device, or completely executed on a remote computing device.

[0159] In conclusion, after reading the detailed disclosure, a person skilled in the art can understand that the detailed disclosure can be presented only in an example manner, and may impose no limitation. Although not expressly stated herein, a person skilled in the art can understand that this specification is intended to include various proper changes, improvements,

and modifications to the embodiments. These changes, improvements, and modifications are intended to be presented in this specification, and fall within the spirit and scope of the example embodiments of this specification.

[0160] Specific embodiments of this specification are described above. Other embodiments
5 fall within the scope of the appended claims. In some cases, the actions or steps described in the claims can be performed in an order different from that in the embodiments, and the desired results can still be achieved. In addition, processes described in the accompanying drawings do not necessarily need a specific order or a sequential order shown to achieve the desired results. In some implementations, multi-tasking and concurrent processing are feasible or may be advantageous.

[0161] In addition, certain terms in this specification are used to describe the embodiments of
10 this specification. For example, "an embodiment", "embodiments", and/or "some embodiments" mean that a particular feature, structure, or characteristic described with reference to the embodiments may be included in at least one embodiment of this specification. Therefore, it can be emphasized and understood that two or more of "embodiments", "one embodiment", or
15 "alternative embodiments" in each part of this specification do not necessarily indicate the same embodiment. In addition, specific features, structures, or features can be properly combined in one or more embodiments of this specification.

[0162] It should be understood that, in the descriptions of the embodiments of this
specification, to help understand a feature, for purposes of simplifying this specification, various
20 features are sometimes combined in a single embodiment, accompanying drawings, or descriptions thereof in this specification. Alternatively, in this specification, various features are distributed in a plurality of embodiments of this specification. However, this does not mean that a combination of these features is necessary. A person skilled in the art may completely extract some features as a separate embodiment for understanding when reading this specification. That is, the
25 embodiments in this specification can also be understood as an integration of a plurality of secondary embodiments. Content of each secondary embodiment is also true when features are less than all features in one of the above-mentioned disclosed embodiments.

CLAIMS

What is claimed is:

1. A data processing method, comprising:

determining at least one table field of a to-be-labeled table and a plurality of seed dimension
5 tables, wherein the seed dimension table comprises an entity association field set including a
plurality of entity association fields, and each of the plurality of entity association fields is used to
associate an entity of the to-be-labeled table with an entity of the seed dimension table;

if an entity association field corresponding to the at least one table field exists in an entity
association field set of at least one seed dimension table, determining labeling information
10 corresponding to data of each row in the to-be-labeled table from the at least one seed dimension
table based on the corresponding entity association field; and

labeling data of each row in the to-be-labeled table based on the labeling information, wherein
the seed dimension table records the labeling information of data of a plurality of rows
corresponding to the entity association fields, and the labeling information comprises information
15 on data holding entity.

2. The method according to claim 1, wherein before the determining at least one table field
of a to-be-labeled table, the method further comprises:

determining an entity model tree of a plurality of data tables in a target database; and
generating, based on the entity model tree, the plurality of seed dimension tables
20 corresponding to the plurality of data tables in the target database, wherein one seed dimension
table corresponds to one type of entity.

3. The method according to claim 1, wherein before the determining labeling information
corresponding to data of each row in the to-be-labeled table, the method further comprises:

determining an entity association field set of each of the plurality of seed dimension tables,
25 wherein the entity association field set comprises a candidate field subset and a lineage field subset.

4. The method according to claim 3, wherein the determining a lineage field subset of each
of the plurality of seed dimension tables comprises:

constructing a lineage graph of a plurality of data tables based on data lineage of each of the
plurality of data tables;
30 determining an associated data table associated with the each of the plurality of seed

dimension tables in the plurality of data tables, wherein the associated data table comprises a reference lineage field corresponding to the entity association field of the each of the plurality of seed dimension tables;

determining a lineage field corresponding to the reference lineage field in a downstream table of each associated data table based on the reference lineage field in each associated data table and the lineage graph; and

determining the lineage field subset of the each of the plurality of seed dimension tables based on the reference lineage field and the lineage field.

5. The method according to claim 4, wherein the determining a lineage field corresponding to the reference lineage field in a downstream table of each associated data table based on the reference lineage field in each associated data table and the lineage graph comprises:

determining an initial lineage field of each data table based on a candidate field set of the plurality of data tables;

determining the downstream table of the associated data table from the plurality of data tables based on a table lineage relationship; and

if the downstream table inherits the reference lineage field in the associated data table, correcting an association relationship of an initial lineage field of the downstream table based on the reference lineage field.

6. The method according to claim 5, wherein the correcting an association relationship of an initial lineage field of the downstream table based on the reference lineage field comprises:

determining an inheritance manner in which the downstream table inherits the reference lineage field in the associated data table; and

correcting the initial lineage field of the downstream table based on the reference lineage field if the inheritance manner is a direct dependency relationship.

7. The method according to claim 1, wherein the method further comprises:

determining an upstream table of the to-be-labeled table based on a table lineage relationship if no entity association field corresponding to the table field exists in an entity association field set of each of the plurality of seed dimension tables; and

performing table labeling on the to-be-labeled table based on labeling information of the upstream table.

8. The method according to claim 1, wherein the method further comprises:

if no entity association field corresponding to the table field exists in an entity association field set of each of the plurality of seed dimension tables, determining whether a derived field in a derived dimension table comprises the table field, wherein the derived dimension table is a data table obtained by associating the seed dimension table with another data table; and

5 if the derived field comprises the table field, determining labeling information corresponding to data of each row of the table field based on an association relationship between the derived dimension table and the seed dimension table.

9. The method according to claim 1, wherein the determining labeling information corresponding to data of each row in the to-be-labeled table based on the entity association field
10 of the at least one seed dimension table comprises:

performing a connection query on data of each row of an entity association field of a target seed dimension table in the at least one seed dimension table and data of each row of the table field; and

determining the labeling information corresponding to data of each row in the to-be-labeled
15 table based on a result of the connection query.

10. The method according to claim 9, wherein the determining the labeling information corresponding to data of each row in the to-be-labeled table based on a result of the connection query comprises:

for covered row data that is in the to-be-labeled table and that is covered by the result of the
20 connection query, determining labeling information of the corresponding covered row data in the to-be-labeled table based on the labeling information of data of each row of the entity association field; and

for uncovered row data that is in the to-be-labeled table and that is not covered by the result of the connection query, labeling the uncovered row data based on a non-standard type of the
25 uncovered row data, wherein the non-standard type comprises a non-standard format and non-standard content.

11. The method according to claim 10, wherein before the labeling the uncovered row data based on a non-standard type of the uncovered row data, the method further comprises:

if a data format of the uncovered row data is not a standard format, determining that the non-
30 standard type of the uncovered row data is a non-standard format; or

if a data format of the uncovered row data is a standard format, and data content of the

uncovered row data is invalid content, determining that the non-standard type of the uncovered row data is non-standard content.

12. The method according to claim 9, wherein before the determining labeling information corresponding to data of each row in the to-be-labeled table, the method further comprises:

5 ranking each seed dimension table based on a priority of each of the at least one seed dimension table; and

determining the target seed dimension table based on a ranking order of the at least one seed dimension table.

13. The method according to claim 1, wherein after the labeling data of each row in the to-be-labeled table, the method further comprises:

performing aggregation processing on the labeling information of data of each row in the to-be-labeled table.

14. The method according to claim 1, wherein before the determining at least one table field of a to-be-labeled table, the method further comprises:

15 determining, based on a preset identification rule, whether the to-be-labeled table needs to be labeled.

15. The method according to any one of claims 1 to 14, wherein the labeling information further comprises at least one of service information or region information.

16. The method according to any one of claims 1 to 14, wherein data holding entities of the plurality of entity association fields are the same and have different names.

17. A data processing device, comprising:

at least one storage medium, comprising at least one instruction set; and

at least one processor, communicatively connected to the at least one storage medium, wherein

25 when the device runs, the at least one processor reads the at least one instruction set, and performs the method according to any one of claims 1 to 16 based on an indication of the at least one instruction set.

18. A non-transitory storage medium, wherein the non-transitory storage medium stores at least one group of executable instructions, and when the at least one group of executable instructions are executed by at least one processor, the method according to any one of claims 1 to 16 is performed.