

SQL 引擎执行安全策略的方法以及 SQL 引擎

本申请要求在 2023 年 12 月 15 日提交国家知识产权局、申请号为 202311733024.3、发明名称为“SQL 引擎的使用方法、计算设备”的中国专利申请的优先权，以及要求在 2024 年 1 月 23 日提交国家知识产权局、申请号为 202410095005.0、发明名称为“SQL 引擎执行安全策略的方法以及 SQL 引擎”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及数据库领域，并且更具体地，涉及一种结构化查询语言 SQL 引擎执行安全策略的方法、SQL 引擎以及计算设备。

背景技术

目前，对于企业级的应用而言，保证数据的安全性和隐私性是及其重要的。数据库系统中的引擎（例如，SQL 引擎）作为数栈底层计算引擎，必须确保数据只能被授权的人员访问，避免出现数据泄露和滥用的情况。具体的，通过给 SQL 引擎配置数据安全策略，使得 SQL 引擎可以根据配置的数据安全策略对数据的安全和隐私进行保护。

相关的技术方案中，对于 SQL 引擎而言，如果该 SQL 引擎使用了某一个提供数据安全策略的软件（例如，Apache Ranger），那么该 SQL 引擎上配置的多种数据安全策略都必须依托于这个软件（例如，Apache Ranger）。使得目前的数据安全体系较为刻板，降低了用户的体验感。

有鉴于此，如何使得数据安全体系变得较为灵活，从而提高用户的体验感成为亟需要解决的技术问题。

发明内容

本申请提供一种 SQL 引擎执行安全策略的方法，该方法可以使得数据安全体系较为灵活，提高用户的体验感。

第一方面，提供了一种 SQL 引擎执行安全策略的方法，该方法应用于数据库系统，该数据库系统中包括数据库和至少一个 SQL 引擎，该至少一个 SQL 引擎用于根据 SQL 语句获取该数据库中存储的数据，该方法包括：每个该 SQL 引擎分别获取第一配置信息，该第一配置信息用于指示该 SQL 引擎上执行的不同类型的安全策略配置各自对应的至少一种软件来源，该不同类型的安全策略分别对应的至少一种软件来源不完全相同；每个该 SQL 引擎根据该第一配置信息，通过执行该第一配置信息中指示的不同软件来源各自所提供的对应的安全策略，对该数据库中存储的数据的安全和/或隐私进行保护。

上述技术方案中，可以使得 SQL 引擎能够灵活地使用不同软件提供的数据安全策略，避免 SQL 引擎上所有类型的数据安全策略与一个软件之间的强绑定关系，使得数据安全体系较为灵活，提高用户的体验感。

结合第一方面，在第一方面的某些实现方式中，每个该 SQL 引擎通过以下中的至少一种方式获取该第一配置信息：每个该 SQL 引擎的接口、配置文件、或可视化配置界面。

结合第一方面，在第一方面的某些实现方式中，该不同类型的安全策略包括第一类型安全策略，该第一类型安全策略对应的软件来源包括第一软件和第二软件，每个该 SQL 引擎在通过该第一软件以及该第二软件各自分别提供的第一类型安全策略的校验的情况下，确定该 SQL 引擎通过该第一类型安全策略的安全校验。

上述技术方案中，可以让多重安全保护同时生效，从而更好的对数据的安全性和/或隐私性进行保护。

结合第一方面，在第一方面的某些实现方式中，该方法还包括：每个该 SQL 引擎分别获取第二配置信息，该第二配置信息用于配置该 SQL 引擎执行该不同类型的安全策略中的部分安全策略或全部安全策略；每个该 SQL 引擎分别根据各自接收到的该第二配置信息执行该不同类型的安全策略中的部分安全策略或全部安全策略。

上述技术方案中，可以使得 SQL 引擎能够灵活的使用多种类型的安全策略中的部分或全部，使得数据安全体系较为灵活，提高用户的体验感。

结合第一方面，在第一方面的某些实现方式中，该不同类型的安全策略包括以下中的至少两种：数据访问权限控制类的安全策略，数据脱敏类的安全策略，数据行级过滤类的安全策略。

结合第一方面，在第一方面的某些实现方式中，该软件来源包括：Ranger，Hive 元数据仓库（Hive meta store，HMS），Hive 访问权限列表 ACL。

结合第一方面，在第一方面的某些实现方式中，该方法应用于云管理平台，该云管理平台用于管理提供云服务的基础设施，该基础设施包括至少一个云数据中心，每个该云数据中心设置有至少一个服务器，该至少一个服务器中运行有该 SQL 引擎。

第二方面，提供了一种 SQL 引擎，该 SQL 引擎位于数据库系统中，该数据库系统中还包括数据库，该 SQL 引擎用于根据 SQL 语句获取该数据库中存储的数据，该 SQL 引擎包括：获取模块，处理模块。其中，获取模块用于获取第一配置信息，该第一配置信息用于指示该 SQL 引擎上执行的不同类型的安全策略配置各自对应的至少一种软件来源，该不同类型的安全策略分别对应的至少一种软件来源不完全相同；处理模块用于根据该第一配置信息，通过执行该第一配置信息中指示的不同软件来源各自所提供的对应的安全策略，对数据库中存储的数据的安全和/或隐私进行保护。

结合第二方面，在第二方面的某些实现方式中，该获取模块具体用于：通过以下中的至少一种方式获取第一配置信息：该 SQL 引擎的接口、配置文件、或可视化配置界面。

结合第二方面，在第二方面的某些实现方式中，该不同类型的安全策略包括第一类型安全策略，该第一类型的安全策略对应的软件来源包括第一软件和第二软件，该处理模块具体用于：在通过该第一软件以及该第二软件各自分别提供的第一类型安全策略的校验的情况下，确定该 SQL 引擎通过该第一类型安全策略的安全校验。

结合第二方面，在第二方面的某些实现方式中，该获取模块，还用于获取第二配置信息，该第二配置信息用于配置该 SQL 引擎执行该不同类型的安全策略中的部分安全策略或全部安全策略；该处理模块，还用于根据该第二配置信息执行该不同类型的安全策略中的部分安全策略或全部安全策略。

结合第二方面，在第二方面的某些实现方式中，该不同类型的安全策略包括以下中的至少两种：数据访问权限控制类的安全策略，数据脱敏类的安全策略，数据行级过滤类的安全策略。

结合第二方面，在第二方面的某些实现方式中，该软件来源包括：Apache Ranger，Hive 元数据仓库 HMS，Hive 访问权限列表 ACL。

结合第二方面，在第二方面的某些实现方式中，该 SQL 引擎运行在至少一个服务器中，该至少一个服务器位于至少一个云数据中心，该至少一个服务器为云管理平台管理的提供云服务的基础设施。

应理解，第二方面及第二方面的各种实现方式中的有益效果请参考第一方面及第一方面的各种实现方式中的有益效果，此处不再赘述。

第三方面，提供了一种数据库系统，该数据库系统包括数据库和至少一个如第二方面以及第二方面中任一种实现方式所提供的 SQL 引擎，至少一个 SQL 引擎用于根据 SQL 语句获取数据库中存储的数据。

第四方面，提供了一种计算设备，包括处理器和存储器，可选地，还包括输入输出接口。其中该处理器用于控制该输入输出接口收发信息，该存储器用于存储计算机程序，该处理器用于从存储器中调用并运行该计算机程序，使得该计算设备执行第一方面或第一方面任意一种可能的实现方式中的方法。

可选地，该处理器可以是通用处理器，可以通过硬件来实现也可以通过软件来实现。当通过硬件实现时，该处理器可以是逻辑电路、集成电路等；当通过软件来实现时，该处理器可以是一个通用处理器，通过读取存储器中存储的软件代码来实现，该存储器可以集成在处理器中，可以位于该处理器之外，独立存在。

第五方面，提供了一种计算设备集群，包括至少一个计算设备，每个计算设备包括处理器和存储器；该至少一个计算设备的处理器用于执行该至少一个计算设备的存储器中存储的指令，以使得该计算设备集群执行第一方面或第一方面任意一种可能的实现方式中的方法。

第六方面，提供了一种芯片，该芯片获取指令并执行该指令来实现上述第一方面以及第一方面的任意一种实现方式中的方法。

可选地，作为一种实现方式，该芯片包括处理器与数据接口，该处理器通过该数据接口读取存储器上存储的指令，执行上述第一方面以及第一方面的任意一种实现方式中的方法。

可选地，作为一种实现方式，该芯片还可以包括存储器，该存储器中存储有指令，该处理器用于执行该存储器上存储的指令，当该指令被执行时，该处理器用于执行第一方面以及第一方面中的任意一种实现方式中的方法。

第七方面，提供了一种包含指令的计算机程序产品，当该指令被计算设备运行时，使得该计算设备执行如上述第一方面以及第一方面的任意一种实现方式中的方法。

第八方面，提供了一种包含指令的计算机程序产品，当该指令被计算设备集群运行时，使得该计算设备集群执行如上述第一方面以及第一方面的任意一种实现方式中的方法。

第九方面，提供了一种计算机可读存储介质，包括计算机程序指令，当该计算机程序指令由计算设备执行时，该计算设备执行如上述第一方面以及第一方面的任意一种实现方式中的方法。

作为示例，这些计算机可读存储包括但不限于如下的一个或者多个：只读存储器（read-only memory, ROM）、可编程 ROM（programmable ROM, PROM）、可擦除的 PROM（erasable PROM, EPROM）、Flash 存储器、电 EPROM（electrically EPROM, EEPROM）以及硬盘驱动器（hard drive）。

可选地，作为一种实现方式，上述存储介质具体可以是非易失性存储介质。

第十方面，提供了一种计算机可读存储介质，包括计算机程序指令，当该计算机程序指令由计算设备集群执行时，该计算设备集群执行如上述第一方面以及第一方面的任意一种实现方式中的方法。

作为示例，这些计算机可读存储包括但不限于如下的一个或者多个：只读存储器（read-only memory, ROM）、可编程 ROM（programmable ROM, PROM）、可擦除的 PROM（erasable PROM, EPROM）、Flash 存储器、电 EPROM（electrically EPROM, EEPROM）以及硬盘驱动器（hard drive）。

可选地，作为一种实现方式，上述存储介质具体可以是非易失性存储介质。

附图说明

图 1 是适用于本申请实施例的一种云场景的示意性框图。

图 2 是本申请实施例提供的一种 SQL 引擎执行安全策略的方法的示意性流程图。

图 3 是本申请实施例提供的一种 SQL 引擎的多种数据安全策略和安全策略来源之间的对应关系。

图 4 是本申请实施例提供的一种为 SQL 引擎配置的不同类别的数据安全策略分别对应的软件来源的示意性框图。

图 5 是本申请实施例提供的一种 SQL 引擎执行安全策略的流程示意图。

图 6 是本申请实施例提供的一种 SQL 引擎 600 的示意性框图。

图 7 是本申请实施例提供的一种数据库系统 700 的示意性框图。

图 8 是本申请实施例提供的一种计算设备 1500 的架构示意图。

图 9 是本申请实施例提供的一种计算设备集群的架构示意图。

图 10 是本申请实施例提供的计算设备 1500A 和 1500B 之间通过网络进行连接的示意图。

具体实施方式

下面将结合附图，对本申请中的技术方案进行描述。

本申请将围绕包括多个设备、组件、模块等的系统来呈现各个方面、实施例或特征。应当理解和明白的是，各个系统可以包括另外的设备、组件、模块等，并且/或者可以并不包括结合附图讨论的所有设备、组件、模块等。此外，还可以使用这些方案的组合。

另外，在本申请实施例中，“示例的”、“例如”等词用于表示作例子、例证或说明。本申请中被描述为“示例”的任何实施例或设计方案不应被解释为比其它实施例或设计方案更优选或更具优势。确切而言，使用示例的一词旨在以具体方式呈现概念。

本申请实施例中，“相应的（corresponding, relevant）”和“对应的（corresponding）”有时可以混用，应当指出的是，在不强调其区别时，其所要表达的含义是一致的。

本申请实施例描述的业务场景是为了更加清楚地说明本申请实施例的技术方案，并不构成对于本申请实施例提供的技术方案的限定，本领域普通技术人员可知，随着网络架构的演变和新业务场景的出现，本申请实施例提供的技术方案对于类似的技术问题，同样适用。

在本说明书中描述的参考“一个实施例”或“一些实施例”等意味着在本申请的一个或多个实施例中包括结合该实施例描述的特定特征、结构或特点。由此，在本说明书中的不同之处出现的语句“在一个实施例

中”、“在一些实施例中”、“在其他一些实施例中”、“在另外一些实施例中”等不是必然都参考相同的实施例，而是意味着“一个或多个但不是所有的实施例”，除非是以其他方式另外特别强调。术语“包括”、“包含”、“具有”及它们的变形都意味着“包括但不限于”，除非是以其他方式另外特别强调。

本申请中，“至少一个”是指一个或者多个，“多个”是指两个或两个以上。“和/或”，描述关联对象的关联关系，表示可以存在三种关系，例如，A 和/或 B，可以表示：包括单独存在 A，同时存在 A 和 B，以及单独存在 B 的情况，其中 A，B 可以是单数或者复数。字符“/”一般表示前后关联对象是一种“或”的关系。“以下至少一项(个)”或其类似表达，是指的这些项中的任意组合，包括单项(个)或复数项(个)的任意组合。例如，a，b，或 c 中的至少一项(个)，可以表示：a，b，c，a-b，a-c，b-c，或 a-b-c，其中 a，b，c 可以是单个，也可以是多个。

为了便于描述，下面先对本申请实施例涉及的概念进行解释说明。

1、结构化查询语言(structured query language, SQL)引擎

SQL 引擎(SQL engine)是用于处理 SQL 查询的软件组件，它是数据库管理系统(database management system, DBMS)的核心组件之一。

下面介绍 SQL 引擎的主要功能。

- 1) 解析 SQL 语句：将 SQL 语句转换为内部表示形式，即解析树；
- 2) 优化查询计划：对解析树进行优化，生成高效的查询执行计划；
- 3) 执行查询计划：根据查询计划执行查询，并从存储在数据库中的数据检索结果。
- 4) 返回结果：将查询结果返回给客户端应用程序。

下面列举几种常见的 SQL 引擎。

1) Hive 引擎，是基于 Hadoop 的一个数据仓库工具，用来进行数据提取、转化、加载，这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。

2) Spark 引擎，是专为大规模数据处理而设计的快速通用的计算引擎。

3) Presto 引擎，是 Facebook 推出的一个开源的分布式 SQL 查询引擎，数据规模可以支持 GB 到 PB 级，主要应用于处理秒级查询的场景。

4) Trino 引擎，是针对 OLAP 设计的用于高效的分布式查询大量数据的分析引擎。Trino 引擎与 Presto 引擎属于同一个开源生态的两个分支。

2、数据安全

数据安全用于对数据库中存储的数据的安全性和隐私性进行保护，避免出现数据泄露和/或数据滥用等情况。具体的，数据安全可以分为引擎层的数据安全和存储层的数据安全。

引擎层的数据安全主要是 SQL 引擎通过执行数据安全策略对数据库中存储的数据的安全性和隐私性进行保护。

下面介绍几种常见的数据安全策略。

1) 数据访问权限控制(data access permission control)类的安全策略

数据访问权限控制类的安全策略是对数据访问的权限进行定义，该定义的维度一般是：数据源/库/表/列(从大到小)。

2) 数据脱敏(data masking)类的安全策略

数据脱敏类的安全策略一般是针对某个用户访问特定表的特定列，来定义对应的数据脱敏策略(可由具体的函数实现)，从而实现同一份数据，不同人看到的内容不同，使得敏感数据对低权限用户不可见。

应理解，数据脱敏是一种保护个人隐私的技术手段，其主要目的是在保证数据使用 and 有效性的同时，对敏感信息进行保密处理，以避免数据泄露带来的安全风险。通过数据脱敏，可以隐藏或改变敏感数据的部分信息，从而降低数据泄露的风险，保护个人隐私和数据安全。数据脱敏的实施需要遵循两个原则：一是尽可能地为脱敏后的应用保留有意义的信息；二是最大程度地防止黑客进行破解。数据脱敏可以分为静态数据脱敏和动态数据脱敏。静态数据脱敏是指将原始数据中的敏感信息进行脱敏处理后，再提供给相关人员使用。这样可以确保在使用数据时不会泄露敏感信息。动态数据脱敏则是在数据传输过程中对敏感信息进行实时脱敏处理，确保数据在传输过程中不会被未经授权的人员获取敏感信息。

3) 数据行级过滤(row level filter)类的安全策略

数据行级过滤类的安全策略是指在数据处理中对每行数据进行筛选，仅保留符合特定条件的数据行，

从而控制用户只能访问数据表的特定数据行。

3、提供数据安全策略的软件

提供数据安全策略的软件也可以称为数据安全策略的来源，其用于向 SQL 引擎提供上述数据安全策略，不同的软件提供的数据安全策略的类型可以相同，或者也可以不同。

下面介绍几种常见的提供数据安全策略的软件。

1) Apache Ranger

Apache Ranger 也可以简称为 Ranger。Apache Ranger 是一个 Hadoop 安全相关的开源组件，它提供了一个非常全面的数据安全管理框架。

Apache Ranger 安全相关的功能十分丰富，管控力度更细。具体的，Apache Ranger 针对 SQL 引擎提供了以下三类不同的数据安全策略：数据访问权限控制类的安全策略、数据脱敏类的安全策略、数据行级过滤类的安全策略。

2) Hive 元数据仓库 (Hive metastore, HMS)

Hive 元数据仓库是 Hive 中存储元数据的地方。Hive 是基于 Hadoop 的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供类 SQL 查询功能。Hive 的元数据包括表、数据库、分区、分桶等信息的描述，这些信息存储在元数据库中。

在 Hive 中，元数据存储在关系数据库管理系统 (relational database management system, RDBMS) 中，例如 MySQL、Derby 等。Hive 内置使用 Derby 来存储元数据，而元数据服务是由 MetaStore 来提供，负责管理客户端对元数据的访问。Hive 的元数据仓库可以存储在 Hadoop 分布式文件系统 (Hadoop distributed file system, HDFS) 上，与数据仓库中的数据和元数据相关联。

在 Hive 中，元数据的管理相对独立于数据的管理，可以通过元数据服务进行访问和操作。通过 Hive 的元数据仓库，可以方便地管理和维护 Hive 中的表、数据库、分区、分桶等元数据信息，保证数据的完整性和一致性。

3) Hive 访问权限列表 (access permission list, ACL)

目前，对于企业级的应用而言，保证数据的安全性和隐私性是及其重要的。数据库系统中的引擎 (例如，SQL 引擎) 作为数栈底层计算引擎，必须确保数据只能被授权的人员访问，避免出现数据泄露和滥用的情况。具体的，通过给 SQL 引擎配置数据安全策略，使得 SQL 引擎可以根据配置的数据安全策略对数据的安全和隐私进行保护。

相关的技术方案中，对于 SQL 引擎而言，如果该 SQL 引擎使用了某一个提供数据安全策略的软件 (例如，Apache Ranger)，那么该 SQL 引擎上配置的多种数据安全策略都必须依托于这个软件 (例如，Apache Ranger)。也即，如果 SQL 引擎使用的数据访问权限控制类的安全策略是 Apache Ranger 提供的，那么，其他安全策略，例如，数据脱敏类的安全策略、数据行级过滤类的安全策略均需要使用 Apache Ranger 提供的。从而造成 SQL 引擎中安全策略与某一个软件 (提供数据安全策略的软件) 之间的强绑定，使得目前的数据安全体系较为刻板，SQL 引擎无法灵活的使用不同软件提供的数据安全策略，降低了用户的体验感。

有鉴于此，本申请实施例提供了一种 SQL 引擎执行安全策略的方法，该方法使得 SQL 引擎可以灵活的使用不同软件提供的数据安全策略，避免 SQL 引擎上所有类型的数据安全策略与一个软件之间的强绑定关系，使得数据安全体系较为灵活，提高用户的体验感。

一种可能的实现方式中，本申请实施例提供的方法可以应用到云服务的场景中。为了便于描述，下面先结合图 1，对云服务的场景进行详细描述。

图 1 是适用于本申请实施例的一种云场景的示意性框图。如图 1 所示，该云场景中可以包括：云管理平台 110、互联网 120 以及客户端 130。

如图 1 所示，云管理平台 110 用于管理提供多个云服务的基础设施。基础设施包括多个云数据中心，每个云数据中心中包括多个服务器，每个服务器中分别包括云服务资源，为租户提供相应的云服务。

具体的，云数据中心中的服务器上运行有 SQL 引擎，由该 SQL 引擎执行本申请实施例提供的方法。

云管理平台 110 可以位于云数据中心，其可以提供访问接口 (如界面或应用程序界面 (application program interface, API))。租户可操作客户端 130 远程接入访问接口在云管理平台 110 注册云账号和密码，并登录云管理平台 110。云管理平台 110 对云账号和密码鉴权成功后，租户可进一步在云管理平台 110 付费选择并购买特定规格 (处理器、内存、磁盘) 的虚拟机。付费购买成功后，云管理平台 110 提供所购

买的虚拟机的远程登录账号密码，客户端 130 可远程登录该虚拟机，在该虚拟机中安装并运行租户的应用。因此，租户可通过云管理平台 110 在云数据中心的创建、管理、登录和操作虚拟机。其中，虚拟机也可称为云服务器（elastic compute service, ECS）、弹性实例（不同的云服务提供商有不同的叫法）。

应理解，云服务的租户可以是个人、企业、学校、医院、行政机关等。

云管理平台 110 的功能包括但不限于用户控制台、计算管理服务、网络管理服务、存储管理服务、鉴权服务、镜像管理服务。用户控制台提供界面或 API 与租户交互，计算管理服务用于管理运行虚拟机和容器的服务器以及裸金属服务器，网络管理服务用于管理网络服务（如网关、防火墙等），存储管理服务用于管理存储服务（如数据桶服务），鉴权服务用于管理租户的账号密码，镜像管理服务用于管理虚拟机镜像。租户使用客户端 130，通过互联网 120 可以登录云管理平台 110，对租用的云服务进行管理。

下面先结合图 2，对本申请实施例提供的一种 SQL 引擎执行安全策略的方法进行详细描述。应理解，图 2 的例子仅仅是为了帮助本领域技术人员理解本申请实施例，而非要将申请实施例限制于图 2 所示例的具体数值或具体场景。本领域技术人员根据图 2 所给出的下面的例子，显然可以进行各种等价的修改或变化，这样的修改和变化也落入本申请实施例的范围内。

图 2 是本申请实施例提供的一种 SQL 引擎执行安全策略的方法的示意性流程图。如图 2 所示，该方法可以包括步骤 210-220，下面分别对步骤 210-220 进行详细描述。

一个示例，该方法应用于数据库系统，该数据库系统中包括数据库和至少一个 SQL 引擎，该至少一个 SQL 引擎用于根据 SQL 语句获取该数据库中存储的数据。

步骤 210：SQL 引擎获取配置信息，该配置信息用于为每种类型的数据安全策略分别配置其所对应的来源软件。

本申请实施例中，可以允许每个 SQL 引擎自由定义其所使用或消费的多种数据安全策略的不同获取来源。也即，可以对每种类型的数据安全策略分别配置其所对应的获取来源。

应理解，每种类型的数据安全策略可以对应一种获取来源，或者也可以对应多种获取来源，本申请对此不做具体限定。

还应理解，不同类型的数据安全策略所对应的获取来源可以包含相同的获取来源，或者也可以不包含相同的获取来源，本申请对此不做具体限定。

上述获取来源也可以称为数据安全策略来源，可以理解为是提供数据安全策略的软件，该提供数据安全策略的软件可以包括但不限于：Apache Ranger、HMS、Hive ACL 等。

需要说明的是，用户在除了部署业务需要的 SQL 引擎外，还需要部署大于 1 种的提供数据安全策略的软件。

上述多种数据安全策略可以包括但不限于：数据访问权限控制类的安全策略、数据脱敏类的安全策略、数据行级过滤类的安全策略等。

可选的，本申请实施例中还可以允许每个 SQL 引擎自由定义是否启用上述多种数据安全策略中的部分安全策略，或全部安全策略。

一种可能的实现方式中，管理员可以对每个 SQL 引擎所使用的多种数据安全策略的不同获取来源进行配置。该配置的实现方式有多种，本申请实施例对此不做具体限定。下面介绍几种可能的实现方式。

1、管理员通过 SQL 引擎的接口对其所使用的多种数据安全策略的不同获取来源进行配置。

具体的，管理员通过 SQL 引擎的应用程序接口（application programming interface, API）对该 SQL 引擎所使用的多种数据安全策略的不同获取来源进行配置。

一个示例，如果 SQL 引擎位于客户端，可以通过该种实现方式对 SQL 引擎进行配置。

2、管理员通过配置文件或可视化的界面对其所使用的多种数据安全策略的不同获取来源进行配置。

一个示例，如果 SQL 引擎位于服务端，可以通过该种实现方式对 SQL 引擎进行配置。

举例说明，如图 3 所示，管理员为数据访问权限控制类的安全策略（也可以简称为 access 配置项）配置的获取来源为安全策略来源 1，为数据脱敏类的安全策略（也可以简称为 masking 配置项）配置的获取来源为安全策略来源 2 以及安全策略来源 3，为数据行级过滤类的安全策略（也可以简称为 row level filter 配置项）配置的获取来源为安全策略来源 3。

本申请实施例中，假设上述安全策略来源 1 为 HMS，安全策略来源 2 为 Apache Ranger，安全策略来源 3 为其他软件。如图 4 所示，通过 SQL 引擎的安全策略配置接口，为数据访问权限控制类的安全策略（也可以简称为 access 配置项）配置的获取来源为 HMS，为数据脱敏类的安全策略（也可以简称为

masking 配置项) 配置的获取来源为 Apache Ranger 以及其他软件, 为数据行级过滤类的安全策略 (也可以简称为 row level filter 配置项) 配置的获取来源为其他软件。

也就是说, SQL 引擎使用的数据访问权限控制类的安全策略是由 HMS 提供的数据访问权限控制类的安全策略, 使用的数据脱敏类的安全策略是由 Apache Ranger 以及其他软件提供的数据脱敏类的安全策略, 使用的数据行级过滤类的安全策略是由其他软件提供的数据行级过滤类的安全策略。

下面以管理员通过配置文件的方式对 SQL 引擎进行配置为例, 结合图 4, 对配置文件的具体实现进行举例说明。

一种可能的实现方式中, 以 SQL 引擎为 Spark 为例, 可以在 Spark 的系统启动配置文件中添加并保存如下参数:

```
spark.ranger.plugin.authorization.enable=false;
spark.HMS.plugin.access.enable=true;
spark.ranger and others.plugin.masking.enable=true;
spark.others.plugin.row_level_filter.enable=true;
```

其中, “spark.ranger.plugin.authorization.enable=false” 表示用于设定关闭 Ranger 提供的数据安全策略; “spark.HMS.plugin.access.enable=true” 表示用于设定启动 HMS 提供的数据访问权限控制类的安全策略 (简称为 access); “spark.ranger and others.plugin.masking.enable=true” 表示用于设定启动 Ranger 以及其他软件提供的数据脱敏类的安全策略 (简称为 masking); “spark.others.plugin.row_level_filter.enable=true” 表示用于设定启动其他软件提供的数据行级过滤类的安全策略 (简称为 row level filter)。

应理解, 上述实现方式中, 假设 SQL 引擎中所有的数据安全策略都使用的是 Ranger 提供的, 因此, 需要先设定关闭 Ranger 提供的数据安全策略, 再设定启动不同来源所提供的数据安全策略。

由于 Spark 有 2 种 SQL 任务提交方式, 所以用户可以选择在以下 3 个不同的位置配置上述参数。

- 1) Spark 的 JDBC 服务端;
- 2) Spark 客户端的本地 conf 文件;
- 3) Spark 的命令行参数。

需要说明的是, 上述是以 SQL 引擎为 Spark 作为示例进行说明的, 对于其他 SQL 引擎均可适用。

另一种可能的实现方式中, 以 SQL 引擎为 Presto 为例, 可以在 Presto 的系统启动配置文件中添加并保存如下参数:

```
security.access.source = HMS;
security.masking.source = Ranger/ others;
security.row_level_filter.source = others;
```

其中, “security.access.source = HMS” 表示用于设定启用 HMS 提供的数据访问权限控制类的安全策略 (简称为 access); “security.masking.source = Ranger/ others” 表示用于设定启用 Ranger 以及其他软件提供的数据脱敏类的安全策略 (简称为 masking); “security.row_level_filter.source = others” 表示用于设定启用其他软件提供的数据行级过滤类的安全策略 (简称为 row level filter)。

需要说明的是, 上述是以 SQL 引擎为 Presto 作为示例进行说明的, 对于其他 SQL 引擎均可适用。

步骤 220: SQL 引擎执行对应软件来源提供的数据安全策略对数据库中存储的数据的安全性和/或隐私性进行保护。

本申请实施例中, SQL 引擎在获取到上述配置信息后, 可以基于该配置信息启用数据安全策略, 实现对数据库中存储的数据的安全性和/或隐私性进行保护。

一种实现方式中, 对于数据访问权限控制类的安全策略而言, SQL 引擎可以在根据 SQL 语句从数据库中获取数据之前, 执行该数据访问权限控制类的安全策略, 对用户访问数据库的权限进行安全校验, 从而实现对数据库中存储的数据的安全性进行保护。

另一种实现方式中, 对于数据脱敏类的安全策略或数据行级过滤类的安全策略, SQL 引擎可以在根据 SQL 语句从数据库中获取数据之后, 执行该数据脱敏类的安全策略或数据行级过滤类的安全策略, 对数据库中存储的数据进行脱敏或行级过滤, 从而实现对数据库中存储的数据的隐私性进行保护。

本申请实施例中, 由于每一种类型的数据安全策略可以来自一种软件, 或者也可以来自两种或两种以上的软件。对于来自两种或两种以上的软件所提供的同一种类型的数据安全策略, 在 SQL 引擎的执行流程中, 需要支持同时让这两种或两种以上的软件所提供的同一种类型的数据安全策略同时生效, 也即

需要确定是否能够通过让这两种或两种以上的软件所提供的同一种类型的数据安全策略的安全校验。这样，可以让多重安全保护同时生效，从而更好的对数据的安全性和/或隐私性进行保护。

例如，以上述的数据脱敏类的安全策略为例，该安全策略既包括 Apache Ranger 提供的数据库脱敏类的安全策略，又包括其他软件提供的数据库脱敏类的安全策略。如图 5 所示，SQL 引擎在执行数据库脱敏类的安全策略时，既要确定是否能够通过 Apache Ranger 提供的数据库脱敏类的安全策略的安全校验，又要确定是否能够通过其他软件提供的数据库脱敏类的安全策略的安全校验。

一种情况，在图 5 中，如果对于数据库脱敏类的安全策略而言，SQL 引擎既通过了 Apache Ranger 提供的数据库脱敏类的安全策略的安全校验，又通过了其他软件提供的数据库脱敏类的安全策略的安全校验，那么可以理解为校验成功。

另一种情况，在图 5 中，如果 SQL 引擎通过了 Apache Ranger 提供的数据库脱敏类的安全策略的安全校验，但是没有通过其他软件提供的数据库脱敏类的安全策略的安全校验，那么可以理解为校验失败。

另一种情况，在图 5 中，如果 SQL 引擎通过了其他软件提供的数据库脱敏类的安全策略的安全校验，但是没有通过 Apache Ranger 提供的数据库脱敏类的安全策略的安全校验，那么也可以理解为校验失败。

另一种情况，在图 5 中，如果 SQL 引擎既没有通过 Apache Ranger 提供的数据库脱敏类的安全策略的安全校验，也没有通过其他软件提供的数据库脱敏类的安全策略的安全校验，那么也可以理解为校验失败。

上述技术方案中，SQL 引擎可以灵活的使用不同软件提供的数据库安全策略，避免 SQL 引擎上所有类型的数据安全策略与一个软件之间的强绑定关系，使得数据安全体系较为灵活，提高用户的体验感。

上文结合图 1 至图 5，详细描述了本申请实施例提供的方法，下面将结合图 6-图 9，详细描述本申请装置的实施例。应理解，方法实施例的描述与装置实施例的描述相互对应，因此，未详细描述的部分可以参见前面方法实施例。

图 6 是本申请实施例提供的一种 SQL 引擎 600 的示意性框图。该 SQL 引擎 600 可以通过软件、硬件或者两者的结合实现。本申请实施例提供的 SQL 引擎 600 可以实现本申请实施例图 2 所示的方法流程，该 SQL 引擎 600 包括：获取模块 610，处理模块 620。其中，获取模块 610 用于获取第一配置信息，该第一配置信息用于指示该 SQL 引擎上执行的不同类型的安全策略配置各自对应的至少一种软件来源，该不同类型的安全策略分别对应的至少一种软件来源不完全相同；处理模块 620 用于根据该第一配置信息，通过执行该第一配置信息中指示的不同软件来源各自所提供的对应的安全策略，对数据库中存储的数据的安全和/或隐私进行保护。

可选的，该获取模块 610 具体用于：通过以下中的至少一种方式获取第一配置信息：该 SQL 引擎的接口、配置文件、或可视化配置界面。

可选的，该不同类型的安全策略包括第一类型安全策略，该第一类型的安全策略对应的软件来源包括第一软件和第二软件，该处理模块 620 具体用于：在通过该第一软件以及该第二软件各自分别提供的第一类型安全策略的校验的情况下，确定该 SQL 引擎通过该第一类型安全策略的安全校验。

可选的，该获取模块 610 还用于获取第二配置信息，该第二配置信息用于配置该 SQL 引擎执行该不同类型的安全策略中的部分安全策略或全部安全策略；该处理模块 620 还用于根据该第二配置信息执行该不同类型的安全策略中的部分安全策略或全部安全策略。

可选的，该不同类型的安全策略包括以下中的至少两种：数据访问权限控制类的安全策略，数据库脱敏类的安全策略，数据行级过滤类的安全策略。

可选的，该软件来源包括：Apache Ranger，Hive 元数据仓库 HMS，Hive 访问权限列表 ACL。

可选的，该 SQL 引擎 600 运行在至少一个服务器中，该至少一个服务器位于至少一个云数据中心，该至少一个服务器为云管理平台管理的提供云服务的基础设施。

这里的 SQL 引擎 600 可以以功能模块的形式体现。这里的术语“模块”可以通过软件和/或硬件形式实现，对此不作具体限定。

例如，“模块”可以是实现上述功能的软件程序、硬件电路或二者结合。示例性的，接下来以获取模块 610 为例，介绍获取模块 610 的实现方式。类似的，其他模块，例如处理模块 620 的实现方式可以参考获取模块 610 的实现方式。

获取模块 610 作为软件功能单元的一种举例，获取模块 610 可以包括运行在计算实例上的代码。其中，计算实例可以包括物理主机（计算设备）、虚拟机、容器中的至少一种。进一步地，上述计算实例可以是一台或者多台。例如，获取模块 610 可以包括运行在多个主机/虚拟机/容器上的代码。需要说明的是，

用于运行该代码的多个主机/虚拟机/容器可以分布在相同的区域（region）中，也可以分布在不同的 region 中。进一步地，用于运行该代码的多个主机/虚拟机/容器可以分布在相同的可用区（availability zone, AZ）中，也可以分布在不同的 AZ 中，每个 AZ 包括一个数据中心或多个地理位置相近的数据中心。其中，通常一个 region 可以包括多个 AZ。

同样，用于运行该代码的多个主机/虚拟机/容器可以分布在同一个虚拟私有云（virtual private cloud, VPC）中，也可以分布在多个 VPC 中。其中，通常一个 VPC 设置在一个 region 内，同一 region 内两个 VPC 之间，以及不同 region 的 VPC 之间跨区通信需在每个 VPC 内设置通信网关，经通信网关实现 VPC 之间的互连。

获取模块 610 作为硬件功能单元的一种举例，获取模块 610 可以包括至少一个计算设备，如服务器等。或者，获取模块 610 也可以是利用专用集成电路（application-specific integrated circuit, ASIC）实现、或可编程逻辑器件（programmable logic device, PLD）实现的设备等。其中，上述 PLD 可以是复杂程序逻辑器件（complex programmable logical device, CPLD）、现场可编程门阵列（field-programmable gate array, FPGA）、通用阵列逻辑（generic array logic, GAL）或其任意组合实现。

获取模块 610 包括的多个计算设备可以分布在相同的 region 中，也可以分布在不同的 region 中。获取模块 610 包括的多个计算设备可以分布在相同的 AZ 中，也可以分布在不同的 AZ 中。同样，获取模块 610 包括的多个计算设备可以分布在同一个 VPC 中，也可以分布在多个 VPC 中。其中，所述多个计算设备可以是服务器、ASIC、PLD、CPLD、FPGA 和 GAL 等计算设备的任意组合。

因此，在本申请的实施例中描述的各示例的模块，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

需要说明的是：上述实施例提供的 SQL 引擎 600 在执行上述方法时，仅以上述各功能模块的划分进行举例说明，实际应用中，可以根据需要而将上述功能分配由不同的功能模块完成，即将 SQL 引擎 600 的内部结构划分成不同的功能模块，以完成以上描述的全部或者部分功能。例如，获取模块 610 可以用于执行上述方法中的任意步骤，处理模块 620 可以用于执行上述方法中的任意步骤。获取模块 610、处理模块 620 负责实现的步骤可以根据需要指定，通过获取模块 610、处理模块 620 分别实现上述方法中不同的步骤来实现上述 SQL 引擎 600 的全部功能。

另外，上述实施例提供的 SQL 引擎 600 与方法实施例属于同一构思，其具体实现过程详见上文中的方法实施例，这里不再赘述。

图 7 是本申请实施例提供的一种数据库系统 700，该数据库系统 700 中包括数据库 710 和至少一个 SQL 引擎 600，该至少一个 SQL 引擎 600 用于在根据 SQL 语句获取该数据库 710 中存储的数据时，执行上述不同类型的安全策略，以便于其对该数据库 710 中存储的数据的安全和/或隐私进行保护。

本申请实施例提供的方法可以由计算设备执行，该计算设备也可以被称为计算机系统。包括硬件层、运行在硬件层之上的操作系统层，以及运行在操作系统层上的应用层。该硬件层包括处理单元、内存和内存控制单元等硬件，随后对该硬件的功能和结构进行详细说明。该操作系统是任意一种或多种通过进程（process）实现业务处理的计算机操作系统，例如，Linux 操作系统、Unix 操作系统、Android 操作系统、iOS 操作系统或 windows 操作系统等。该应用层包含浏览器、通讯录、文字处理软件、即时通信软件等应用程序。并且，可选地，该计算机系统是智能手机等手持设备，或个人计算机等终端设备，本申请并未特别限定，只要能够通过本申请实施例提供的方法即可。本申请实施例提供的方法的执行主体可以是计算设备，或者，是计算设备中能够调用程序并执行程序的功能模块。

下面结合图 8，对本申请实施例提供的一种计算设备进行详细描述。

图 8 是本申请实施例提供的一种计算设备 1500 的架构示意图。该计算设备 1500 可以是服务器或者计算机或者其他具有计算能力的设备。图 8 所示的计算设备 1500 包括：至少一个处理器 1510 和存储器 1520。

应理解，本申请不限定计算设备 1500 中的处理器、存储器的个数。

处理器 1510 执行存储器 1520 中的指令，使得计算设备 1500 实现本申请提供的方法。或者，处理器 1510 执行存储器 1520 中的指令，使得计算设备 1500 实现本申请提供的各功能模块，从而实现本申请提供的方法。

可选地，计算设备 1500 还包括通信接口 1530。通信接口 1530 使用例如但不限于网络接口卡、收发器一类的收发模块，来实现计算设备 1500 与其他设备或通信网络之间的通信。

可选地，计算设备 1500 还包括系统总线 1540，其中，处理器 1510、存储器 1520 和通信接口 1530 分别与系统总线 1540 连接。处理器 1510 能够通过系统总线 1540 访问存储器 1520，例如，处理器 1510 能够通过系统总线 1540 在存储器 1520 中进行数据读写或代码执行。该系统总线 1540 是快捷外设部件互连标准（peripheral component interconnect express, PCI）总线或扩展工业标准结构（extended industry standard architecture, EISA）总线等。所述系统总线 1540 分为地址总线、数据总线、控制总线等。为便于表示，图 8 中仅用一条粗线表示，但并不表示仅有一根总线或一种类型的总线。

一种可能的实现方式，处理器 1510 的功能主要是解释计算机程序的指令（或者说，代码）以及处理计算机软件中的数据。其中，该计算机程序的指令以及计算机软件中的数据能够保存在存储器 1520 或者缓存 1516 中。

可选地，处理器 1510 可能是集成电路芯片，具有信号的处理能力。作为示例而非限定，处理器 1510 是通用处理器、数字信号处理器（digital signal processor, DSP）、专用集成电路（application specific integrated circuit, ASIC）、现成可编程门阵列（field programmable gate array, FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。其中，通用处理器是微处理器等。例如，该处理器 1510 是中央处理单元（central processing unit, CPU）。

可选地，每个处理器 1510 包括至少一个处理单元 1512 和内存控制单元 1514。

可选地，处理单元 1512 也称为核心（core）或内核，是处理器最重要的组成部分。处理单元 1512 是由单晶硅以一定的生产工艺制造出来的，处理器所有的计算、接受命令、存储命令、处理数据都由核心执行。处理单元分别独立地运行程序指令，利用并行计算的能力加快程序的运行速度。各种处理单元都具有固定的逻辑结构，例如，处理单元包括例如，一级缓存、二级缓存、执行单元、指令级单元和总线接口等逻辑单元。

一种实现举例，内存控制单元 1514 用于控制存储器 1520 与处理单元 1512 之间的数据交互。具体地说，内存控制单元 1514 从处理单元 1512 接收内存访问请求，并基于该内存访问请求控制针对内存的访问。作为示例而非限定，内存控制单元是内存管理单元（memory management unit, MMU）等器件。

一种实现举例，各内存控制单元 1514 通过系统总线进行针对存储器 1520 的寻址。并且在系统总线中配置仲裁器（图 8 中未示出），该仲裁器负责处理和协调多个处理单元 1512 的竞争访问。

一种实现举例，处理单元 1512 和内存控制单元 1514 通过芯片内部的连接线，例如地址线，通信连接，从而实现处理单元 1512 和内存控制单元 1514 之间的通信。

可选地，每个处理器 1510 还包括缓存 1516，其中，缓存是数据交换的缓冲区（称作 cache）。当处理单元 1512 要读取数据时，会首先从缓存中查找需要的数据，如果找到了则直接执行，找不到的话则从存储器中找。由于缓存的运行速度比存储器快得多，故缓存的作用就是帮助处理单元 1512 更快地运行。

存储器 1520 能够为计算设备 1500 中的进程提供运行空间，例如，存储器 1520 中保存用于生成进程的计算机程序（具体地说，是程序的代码）。计算机程序被处理器运行而生成进程后，处理器在存储器 1520 中为该进程分配对应的存储空间。进一步的，上述存储空间进一步包括文本段、初始化数据段、位初始化数据段、栈段、堆段等等。存储器 1520 在上述进程对应的存储空间中保存进程运行期间产生的数据，例如，中间数据，或过程数据等等。

可选地，存储器也称为内存，其作用是用于暂时存放处理器 1510 中的运算数据，以及与硬盘等外部存储器交换的数据。只要计算机在运行中，处理器 1510 就会把需要运算的数据调到内存中进行运算，当运算完成后处理单元 1512 再将结果传送出来。

作为示例而非限定，存储器 1520 是易失性存储器或非易失性存储器，或可包括易失性和非易失性存储器两者。其中，非易失性存储器是只读存储器（read-only memory, ROM）、可编程只读存储器（programmable ROM, PROM）、可擦除可编程只读存储器（erasable PROM, EPROM）、电可擦除可编程只读存储器（electrically EPROM, EEPROM）或闪存。易失性存储器是随机存取存储器（random access memory, RAM），其用作外部高速缓存。通过示例性但不是限制性说明，许多形式的 RAM 可用，例如静态随机存取存储器（static RAM, SRAM）、动态随机存取存储器（dynamic RAM, DRAM）、同步动态随机存取存储器（synchronous DRAM, SDRAM）、双倍数据速率同步动态随机存取存储器（double data rate SDRAM, DDR SDRAM）、增强型同步动态随机存取存储器（enhanced SDRAM, ESDRAM）、同

步连接动态随机存取存储器 (synchlink DRAM, SLD RAM) 和直接内存总线随机存取存储器 (direct rambus RAM, DR RAM)。应注意, 本文描述的系统和方法的存储器 1520 旨在包括但不限于这些和任意其它适合类型的存储器。

以上列举的计算设备 1500 的结构仅为示例性说明, 本申请并未限定于此, 本申请实施例的计算设备 1500 包括现有技术中计算机系统种的各种硬件, 例如, 计算设备 1500 还包括除存储器 1520 以外的其他存储器, 例如, 磁盘存储器等。本领域的技术人员应当理解, 计算设备 1500 还可以包括实现正常运行所必须的其他器件。同时, 根据具体需要, 本领域的技术人员应当理解, 上述计算设备 1500 还可包括实现其他附加功能的硬件器件。此外, 本领域的技术人员应当理解, 上述计算设备 1500 也可仅仅包括实现本申请实施例所必须的器件, 而不必包括图 8 中所示的全部器件。

本申请实施例还提供了一种计算设备集群。该计算设备集群包括至少一台计算设备。该计算设备可以是服务器。在一些实施例中, 计算设备也可以是台式机、笔记本电脑或者智能手机等终端设备。

如图 9 所示, 所述计算设备集群包括至少一个计算设备 1500。计算设备集群中的一个或多个计算设备 1500 中的存储器 1520 中可以存有相同的用于执行上述方法的指令。

在一些可能的实现方式中, 该计算设备集群中的一个或多个计算设备 1500 中的存储器 1520 也可以分别存有用于执行上述方法的部分指令。换言之, 一个或多个计算设备 1500 的组合可以共同执行上述方法的指令。

需要说明的是, 计算设备集群中的不同的计算设备 1500 中的存储器 1520 可以存储不同的指令, 分别用于执行上述 SQL 引擎的部分功能。也即, 不同的计算设备 1500 中的存储器 1520 存储的指令可以实现将上述 SQL 引擎中的一个或多个模块的功能。

在一些可能的实现方式中, 计算设备集群中的一个或多个计算设备可以通过网络连接。其中, 所述网络可以是广域网或局域网等等。图 10 示出了一种可能的实现方式。如图 10 所示, 两个计算设备 1500A 和 1500B 之间通过网络进行连接。具体地, 通过各个计算设备中的通信接口与所述网络进行连接。

应理解, 图 10 中示出的计算设备 1500A 的功能也可以由多个计算设备 1500 完成。同样, 计算设备 1500B 的功能也可以由多个计算设备 1500 完成。

本实施例中, 还提供了一种包含指令的计算机程序产品, 所述计算机程序产品可以是包含指令的, 能够运行在计算设备上或被储存在任何可用介质中的软件或程序产品。当其在计算设备上运行时, 使得计算设备执行上述所提供的方法, 或者使得该计算设备实现上述提供的 SQL 引擎的功能。

本实施例中, 还提供了一种计算机可读存储介质, 计算机可读存储介质可以是计算设备能够存储的任何可用介质或者是包含一个或多个可用介质的数据中心等数据存储设备。所述可用介质可以是磁性介质, (例如, 软盘、硬盘、磁带)、光介质 (例如, DVD)、或者半导体介质 (例如固态硬盘) 等。该计算机可读存储介质包括指令, 当计算机可读存储介质中的指令在计算设备上被执行时, 使得计算设备执行上述所提供的方法。

应理解, 在本申请的各种实施例中, 上述各过程的序号的大小并不意味着执行顺序的先后, 各过程的执行顺序应以其功能和内在逻辑确定, 而不应对本申请实施例的实施过程构成任何限定。

本领域普通技术人员可以意识到, 结合本文中所公开的实施例描述的各示例的单元及算法步骤, 能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行, 取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能, 但是这种实现不应认为超出本申请的范围。

所属领域的技术人员可以清楚地了解到, 为描述的方便和简洁, 上述描述的系统、装置和单元的具体工作过程, 可以参考前述方法实施例中的对应过程, 在此不再赘述。

在本申请所提供的几个实施例中, 应该理解到, 所揭露的系统、SQL 引擎和方法, 可以通过其它的方式实现。例如, 以上所描述的装置实施例仅仅是示意性的, 例如, 所述单元的划分, 仅仅为一种逻辑功能划分, 实际实现时可以有另外的划分方式, 例如多个单元或组件可以结合或者可以集成到另一个系统, 或一些特征可以忽略, 或不执行。另一点, 所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口, 装置或单元的间接耦合或通信连接, 可以是电性, 机械或其它的形式。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的, 作为单元显示的部件可以是或者也可以不是物理单元, 即可以位于一个地方, 或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。

所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用，可以存储在一个计算机可读存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（read-only memory, ROM）、随机存取存储器（random access memory, RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以所述权利要求的保护范围为准。

权 利 要 求 书

1. 一种结构化查询语言 SQL 引擎执行安全策略的方法，其特征在于，所述方法应用于数据库系统，所述数据库系统中包括数据库和至少一个 SQL 引擎，所述至少一个 SQL 引擎用于根据 SQL 语句获取所述数据库中存储的数据，所述方法包括：

每个所述 SQL 引擎分别获取第一配置信息，所述第一配置信息用于指示所述 SQL 引擎上执行的不同类型的安全策略配置各自对应的至少一种软件来源，所述不同类型的安全策略分别对应的至少一种软件来源不完全相同；

每个所述 SQL 引擎根据所述第一配置信息，通过执行所述第一配置信息中指示的不同软件来源各自所提供的对应的安全策略，对所述数据库中存储的数据的安全和/或隐私进行保护。

2. 根据权利要求 1 所述的方法，其特征在于，所述每个所述 SQL 引擎获取所述第一配置信息，包括：

每个所述 SQL 引擎通过以下中的至少一种方式获取所述第一配置信息：每个所述 SQL 引擎的接口、配置文件、或可视化配置界面。

3. 根据权利要求 1 或 2 所述的方法，其特征在于，所述不同类型的安全策略包括第一类型安全策略，所述第一类型安全策略对应的软件来源包括第一软件和第二软件，

所述每个所述 SQL 引擎根据所述第一配置信息，通过执行所述第一配置信息中指示的不同软件来源各自所提供的对应的安全策略，对所述数据库中存储的数据的安全和/或隐私进行保护，包括：

每个所述 SQL 引擎在通过所述第一软件以及所述第二软件各自分别提供的第一类型安全策略的校验的情况下，确定所述 SQL 引擎通过所述第一类型安全策略的安全校验。

4. 根据权利要求 1 至 3 中任一项所述的方法，其特征在于，所述方法还包括：

每个所述 SQL 引擎分别获取第二配置信息，所述第二配置信息用于配置所述 SQL 引擎执行所述不同类型的安全策略中的部分安全策略或全部安全策略；

每个所述 SQL 引擎分别根据各自接收到的所述第二配置信息执行所述不同类型的安全策略中的部分安全策略或全部安全策略。

5. 根据权利要求 1 至 4 中任一项所述的方法，其特征在于，所述不同类型的安全策略包括以下中的至少两种：数据访问权限控制类的安全策略，数据脱敏类的安全策略，数据行级过滤类的安全策略。

6. 根据权利要求 1 至 5 中任一项所述的方法，其特征在于，所述软件来源包括：Apache Ranger，Hive 元数据仓库 HMS，Hive 访问权限列表 ACL。

7. 根据权利要求 1 至 6 中任一项所述的方法，其特征在于，所述方法应用于云管理平台，所述云管理平台用于管理提供云服务的基础设施，所述基础设施包括至少一个云数据中心，每个所述云数据中心设置有至少一个服务器，所述至少一个服务器中运行有所述 SQL 引擎。

8. 一种结构化查询语言 SQL 引擎，其特征在于，所述 SQL 引擎位于数据库系统中，所述数据库系统中还包括数据库，所述 SQL 引擎用于根据 SQL 语句获取所述数据库中存储的数据，所述 SQL 引擎包括：

获取模块，用于获取第一配置信息，所述第一配置信息用于指示所述 SQL 引擎上执行的不同类型的安全策略配置各自对应的至少一种软件来源，所述不同类型的安全策略分别对应的至少一种软件来源不完全相同；

处理模块，用于根据所述第一配置信息，通过执行所述第一配置信息中指示的不同软件来源各自所提供的对应的安全策略，对所述数据库中存储的数据的安全和/或隐私进行保护。

9. 根据权利要求 8 所述的 SQL 引擎，其特征在于，所述获取模块具体用于：

通过以下中的至少一种方式获取所述第一配置信息：所述 SQL 引擎的接口、配置文件、或可视化配置界面。

10. 根据权利要求 8 或 9 所述的 SQL 引擎，其特征在于，所述不同类型的安全策略包括第一类型安全策略，所述第一类型的安全策略对应的软件来源包括第一软件和第二软件，

所述处理模块具体用于：

在通过所述第一软件以及所述第二软件各自分别提供的第一类型安全策略的校验的情况下，确定所述 SQL 引擎通过所述第一类型安全策略的安全校验。

11. 根据权利要求 8 至 10 中任一项所述的 SQL 引擎，其特征在于，

所述获取模块，还用于获取第二配置信息，所述第二配置信息用于配置所述 SQL 引擎执行所述不同类型的安全策略中的部分安全策略或全部安全策略；

所述处理模块，还用于根据所述第二配置信息执行所述不同类型的安全策略中的部分安全策略或全部安全策略。

12. 根据权利要求 8 至 11 中任一项所述的 SQL 引擎，其特征在于，所述不同类型的安全策略包括以下中的至少两种：数据访问权限控制类的安全策略，数据脱敏类的安全策略，数据行级过滤类的安全策略。

13. 根据权利要求 8 至 12 中任一项所述的 SQL 引擎，其特征在于，所述软件来源包括：Apache Ranger，Hive 元数据仓库 HMS，Hive 访问权限列表 ACL。

14. 根据权利要求 8 至 13 中任一项所述的 SQL 引擎，其特征在于，所述 SQL 引擎运行在至少一个服务器中，所述至少一个服务器位于至少一个云数据中心，所述至少一个服务器为云管理平台管理的提供云服务的基础设施。

15. 一种数据库系统，其特征在于，所述数据库系统包括数据库和至少一个如权利要求 8 至 14 中任一项所述的 SQL 引擎，至少一个所述 SQL 引擎用于根据 SQL 语句获取所述数据库中存储的数据。

16. 一种计算设备集群，其特征在于，包括至少一个计算设备，每个计算设备包括处理器和存储器；所述至少一个计算设备的处理器用于执行所述至少一个计算设备的存储器中存储的指令，以使得所述计算设备集群执行如权利要求 1 至 7 中任一项所述的方法。

17. 一种包含指令的计算机程序产品，其特征在于，当所述指令被计算设备集群运行时，使得所述计算设备集群执行如权利要求的 1 至 7 中任一项所述的方法。

18. 一种计算机可读存储介质，其特征在于，包括计算机程序指令，当所述计算机程序指令由计算设备集群执行时，所述计算设备集群执行如权利要求 1 至 7 中任一项所述的方法。