(19) Deutsches Patent- und Markenamt

(10) **DE 20 2019 005 594 U1** 2021.03.18

(12) # Gebrauchsmusterschrift

(21) Aktenzeichen: **20 2019 005 594.0**
(22) Anmeldetag: **06.08.2019**
(67) aus Patentanmeldung: **EP 19 84 8210.1**
(47) Eintragungstag: **08.02.2021**
(45) Bekanntmachungstag im Patentblatt: **18.03.2021**

(51) Int Cl.: ***G06F 16/2455*** (2019.01)
G06F 16/21 (2019.01)
G06F 21/62 (2013.01)

(30) Unionspriorität:
16/055,824 **06.08.2018** **US**

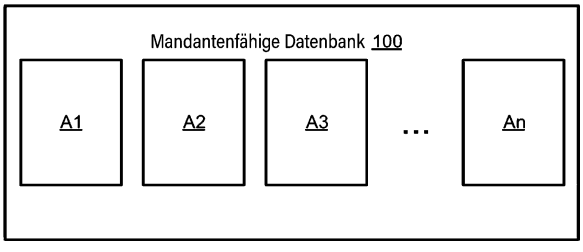
(73) Name und Wohnsitz des Inhabers:
Snowflake Inc., San Mateo, CA, US

(74) Name und Wohnsitz des Vertreters:
Betten & Resch Patent- und Rechtsanwälte PartGmbB, 80333 München, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Sicheres gemeinsames Benutzen von Daten in einem mandantenfähigen Datenbanksystem**

(57) Hauptanspruch: System, das umfasst: einen Speicher; und einen oder mehrere Prozessoren, die betriebstechnisch an den Speicher gekoppelt sind, wobei der eine oder die mehreren Prozessoren:
Sichtprivilegien einer sicheren Sicht einer oder mehreren von mehreren zugrundeliegenden Einzelheiten eines Mitbenutzungsobjekts eines ersten Kontos zuzuordnen, so dass jede der einen oder der mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts eine Definition der sicheren Sicht umfasst;
einem zweiten Konto kontoübergreifende Zugriffsrechte auf das Mitbenutzungsobjekt des ersten Kontos zu gewähren;
in Reaktion auf das Empfangen einer Anforderung vom zweiten Konto, auf das Mitbenutzungsobjekt zuzugreifen, das Modifizieren eines Abfrageplans der Anforderung unter Verwendung eines sicheren Projektionsoperators, um sicherzustellen, dass alle sicherheitsbezogenen Filterungen, die die Anforderung betreffen, stattfinden, bevor irgendein Ausdruck, der die Anforderung betrifft und sich außerhalb der sicheren Sicht befindet, ausgewertet wird, wobei die Ausdrücke, die nicht von der sicheren Sicht stammen, nicht unter eine Grenze der sicheren Sicht nach unten geschoben werden können, so dass die eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts vor dem zweiten Konto verborgen und für das erste Konto sichtbar sind; und
durch das zweite Konto auf das Mitbenutzungsobjekt zuzugreifen, ohne eine Kopie von irgendeiner der einen ...



Beschreibung

VERWANDTE ANMELDUNGEN

[0001] Diese Anmeldung ist eine Fortsetzung der U.S.-Patentanmeldung, laufende Nr. 16/694.906, die eine Fortsetzung der U.S.-Patentanmeldung, laufende Nr. 16/055.824, eingereicht am 06. August 2018, mit dem Titel „SECURE DATA SHA-RING IN A MULTI-TENANT DATABASE SYSTEM“, ist, deren gesamte Inhalte durch Bezugnahme hier mit aufgenommen sind.

TECHNISCHES GEBIET

[0002] Die vorliegende Offenbarung bezieht sich auf Datenbanken und bezieht sich insbesondere auf sichere Sichten für die gemeinsame Benutzung von Daten ohne Kopieren in einem mandantenfähigen Datenbanksystem.

[0003] In Übereinstimmung mit den Anforderungen des Gebrauchsmustergesetzes sind nur Vorrichtungen, wie sie in den beigefügten Ansprüchen definiert sind, aber keine Verfahren als das Gebrauchsmuster zu schützen und der Gegenstand des Gebrauchsmusters. Falls in der Beschreibung auf Verfahren Bezug genommen wird, dienen diese Bezugnahmen lediglich dazu, die Vorrichtung oder die Vorrichtungen, für die mit den beigefügten Ansprüchen Schutz gesucht wird, zu veranschaulichen.

HINTERGRUND

[0004] Datenbanken werden für die Datenspeicherung und den Datenzugriff in Rechenanwendungen weitgehend verwendet. Die Datenbanken können eine oder mehrere Tabellen enthalten, die Daten enthalten oder auf Daten verweisen, die unter Verwendung von Abfragen gelesen, modifiziert oder gelöscht werden können. Die Datenbanken können zum Speichern von und/oder Zugreifen auf persönliche Informationen oder andere sensible Informationen verwendet werden. Die sichere Speicherung und der sichere Zugriff auf Datenbankdaten kann durch das Verschlüsseln und/oder Speichern von Daten in einer verschlüsselten Form bereitgestellt werden, um unberechtigten Zugriff zu verhindern. In einigen Fällen kann das gemeinsame Benutzen von Daten wünschenswert sein, um andere Parteien Abfragen gegen einen Datensatz ausführen zu lassen.

[0005] In Datenbanksystemen werden die Sichten häufig als Sicherheitsmechanismen verwendet, durch die ein Konto den Zugriff auf bestimmte Felder von Tabellen einschränken kann, um Informationen zu verbergen. Eine Sicht erlaubt, dass auf das Ergebnis einer Abfrage zugegriffen wird, als wäre es eine Tabelle. Irgendein Abfrageausdruck, der ein gültiges Ergebnis zurückgibt, kann verwendet werden, um eine Sicht zu erzeugen, einschließlich des Auswählens von Spalten in einer Tabelle, des Auswählens eines spezifischen Datenbereichs in Tabellenspalten und des Verknüpfens von Daten aus zwei oder mehr Tabellen. Die Sichten können verwendet werden, um Abfragen zu vereinfachen und/oder Daten zu verbergen.

Figurenliste

[0006] Nicht einschränkende und nicht vollständige Implementierungen der vorliegenden Offenbarung werden bezüglich der folgenden Figuren beschrieben, wobei sich überall in den verschiedenen Ansichten gleiche Bezugszeichen auf gleiche oder ähnliche Teile beziehen, wenn es nicht anderweitig spezifiziert ist. Die Vorteile der vorliegenden Offenbarung werden bezüglich der folgenden Beschreibung und der beigefügten Zeichnungen besser verstanden, wobei:

Fig. 1 ein schematischer Blockschaltplan ist, der Konten in einer mandantenfähigen Datenbank gemäß einer Ausführungsform veranschaulicht;

Fig. 2 eine schematische graphische Darstellung ist, die ein System zum Bereitstellen von und zum Zugreifen auf Datenbankdienste gemäß einer Ausführungsform veranschaulicht;

Fig. 3 eine schematische graphische Darstellung ist, die eine mandantenfähige Datenbank mit der Trennung von Speicher- und Rechenbetriebsmitteln gemäß einer Ausführungsform veranschaulicht;

Fig. 4 ein schematischer Blockschaltplan ist, der Objekthierarchien gemäß einer Ausführungsform veranschaulicht;

Fig. 5 eine schematische graphische Darstellung ist, die den rollenbasierten Zugriff gemäß einer Ausführungsform veranschaulicht;

Fig. 6 eine schematische graphische Darstellung ist, die eine Benutzungsbewilligung zwischen den Rollen gemäß einer Ausführungsform veranschaulicht;

Fig. 7 eine schematische graphische Darstellung ist, die ein Mitbenutzungsobjekt gemäß einer Ausführungsform veranschaulicht;

Fig. 8 eine schematische graphische Darstellung ist, die kontenübergreifende Bewilligungen gemäß einer Ausführungsform veranschaulicht;

Fig. 9 ein schematischer Blockschaltplan ist, der die Komponenten einer Mitbenutzungskomponente gemäß einer Ausführungsform veranschaulicht;

Fig. 10 ein schematischer Ablaufplan ist, der ein Verfahren zum Verwenden sicherer Sichten, wenn Datenbankdaten zwischen Konten ge-

meinsam benutzt werden, gemäß einer Ausführungsform veranschaulicht;

Fig. 11 ein schematischer Blockschaltplan ist, der einen Prozessablauf für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion gemäß einer Ausführungsform veranschaulicht;

Fig. 12 ein schematischer Blockschaltplan ist, der einen Prozessablauf für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion gemäß einer Ausführungsform veranschaulicht;

Fig. 13 ein schematischer Blockschaltplan ist, der einen Prozessablauf für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion gemäß einer Ausführungsform veranschaulicht;

Fig. 14 ein schematischer Blockschaltplan ist, der einen Prozessablauf für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion gemäß einer Ausführungsform veranschaulicht;

Fig. 15 ein schematischer Blockschaltplan ist, der einen Prozessablauf für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion gemäß einer Ausführungsform veranschaulicht;

Fig. 16 ein schematischer Blockschaltplan ist, der einen Prozessablauf für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion gemäß einer Ausführungsform veranschaulicht;

Fig. 17 ein schematischer Blockschaltplan ist, der einen Prozessablauf für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion gemäß einer Ausführungsform veranschaulicht;

Fig. 18 ein schematischer Blockschaltplan ist, der einen Prozessablauf für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion gemäß einer Ausführungsform veranschaulicht;

Fig. 19 ein schematischer Blockschaltplan ist, der einen Prozessablauf für eine Abfrage, wie sie in einem Abfrageprofil erscheint, gemäß einer Ausführungsform veranschaulicht; und

Fig. 20 ein Blockschaltplan ist, der eine beispielhafte Rechenvorrichtung oder ein beispielhaftes Rechensystem darstellt, die bzw. das mit einer oder mehreren hier offenbarten Ausführungsformen konsistent ist.

AUSFÜHRLICHE BESCHREIBUNG

[0007] In den Datenbanksystemen werden die Sichten im Allgemeinen als Sicherheitsmechanismen verwendet, durch die ein Konto den Zugriff auf bestimmte Felder von Tabellen einschränken kann, um Informationen zu verbergen. Die Implementierung von Sichten und wie die Implementierung von Sichten gehandhabt wird, kann potentiell zum Entweichen von Informationen führen. Während der Abfrageoptimierung können z. B. bestimmte Filter über die Sichtdefinition näher zu den Eingabetabellen geschoben werden, wobei die Informationen einem Anwender verraten werden können, wenn ein anwenderspezifizierter Filter ausgewertet wird, bevor sichere Prädikate ausgewertet werden.

[0008] Eine Sicht erlaubt, dass auf das Ergebnis einer Abfrage zugegriffen wird, als ob es eine Tabelle wäre. Irgendein Abfrageausdruck, der ein gültiges Ergebnis zurückgibt, kann benutzt werden, um eine Sicht zu erzeugen, einschließlich des Auswählens von Spalten in einer Tabelle, des Auswählens eines spezifischen Datenbereichs in den Tabellenspalten und des Verknüpfens von Daten aus zwei oder mehr Tabellen. Die Sichten können benutzt werden, um Abfragen zu vereinfachen und/oder Daten zu verbergen. Zusätzlich können die Sichten Abfrageleistungsvorteile aufweisen. Der Anmelder erkennt, dass, wenn in einer Abfrage auf eine Sicht verwiesen wird, die Abfrage und die Sicht zusammen verarbeitet werden können, so dass die Sicht effizienter ausgewertet werden kann, als wenn sie separat ausgewertet würde.

[0009] Bezüglich der sicheren Sichten erfordern einige Optimierungen für die Sichten Zugriff auf die zugrundeliegenden Daten in den Basistabellen für die Sicht. Der Zugriff auf die zugrundeliegenden Daten kann es ermöglichen, dass die Daten, die vor den Anwendern der Sichten verborgen sind, z. B. durch anwenderdefinierte Funktionen (UDF) oder ein anderes programmatisches Verfahren aufgedeckt werden. Die sicheren Sichten, wie sie in der vorliegenden Anwendung offenbart werden, verwenden derartige Optimierungen nicht und stellen sicher, dass die Anwender keinen Zugriff auf die zugrundeliegenden Daten haben. Gemäß einer Ausführungsform kann der Abfrageausdruck, um eine Standardansicht (die außerdem als die Sichtdefinition oder der Sichttext bezeichnet werden kann) zu erzeugen, für die Anwender sichtbar sein. In einer derartigen Ausführungsform kann ein Anwender aufgrund von Sicherheits- oder Geheimhaltungsgründen nicht wünschen, die zugrundeliegenden Tabellen oder internen Struktureinzelheiten für eine Sicht aufzudecken. Bei einer sicheren Sicht, wie sie in der vorliegenden Anwendung offenbart wird, sind die Sichtdefinition und die Einzelheiten nur für autorisierte Anwender, d. h., für Anwen-

der, denen die Rolle bewilligt worden ist, der die Sicht gehört, sichtbar.

[0010] Eine Sicht kann als sichere Sicht definiert sein, wenn sie für die Datengeheimhaltung spezifisch bestimmt ist, oder um den Zugriff auf die Daten einzuschränken, die nicht für alle Konten der zugrundeliegenden Tabelle aufgedeckt sein sollten. In einer nicht sicheren Sicht könnten Daten aufgedeckt werden, wenn ein Konto z. B. nur auf eine Teilmenge von Daten Zugriff hat. Ein beispielhaftes Szenario, bei dem Informationen verraten werden können, enthält das Folgende. Eine Datenbanktabelle enthält Mitarbeiterdaten für ein gesamtes Unternehmen, wobei eine Sicht definiert ist, um die Mitarbeiterdaten für eine einzelne Abteilung aufzudecken. Einem Konto, das z. B. zu einem Leiter der einzelnen Abteilung gehört, wird der Zugriff auf die Sicht, nicht aber auf die zugrundeliegende Tabelle bewilligt. In einer derartigen Ausführungsform kann ein Anwender Einblick in Daten aus anderen Abteilungen gewinnen, indem er in der Sicht abfragt. Falls der Filter in der Abfrage vor dem Verknüpfen zwischen dem „Mitarbeiter"-Feld und dem „Abteilungs"-Feld ausgewertet wird, kann der Anwender ableiten, ob irgendjemand im gesamten Unternehmen eine bestimmte Metrik erfüllt (z. B. ein Gehalt über einem x Betrag erhält, usw.). Eine derartige Technik kann verwendet werden, um irgendwelche Daten innerhalb der Basistabelle abzufragen. Eine anwenderdefinierte Funktion, die einen Zustand über die Aufrufe überträgt, kann in einer ähnlichen Weise benutzt werden, um die Ergebnisse noch raffinierterer Abfragen an die Daten der Basistabelle zu verraten.

[0011] Die sicheren Sichten erlauben es Datenbank-Klientenkonten, einen eingeschränkten Datensatz anderen Anwendern ohne die Möglichkeit aufzudecken, dass die zugrundeliegenden, uneingeschränkten Daten diesen anderen Anwendern aufgedeckt werden. Ähnlich zu den Sichten ermöglichen es die anwenderdefinierten Funktionen (UDFs) mit zusätzlichen Fähigkeiten aufgrund der Argumentverarbeitung und der Unterstützung prozeduraler Sprachen einem Klientenkonto, eingeschränkte Daten für andere Anwender aufzudecken. Die sicheren UDFs können die gleichen Sicherheitsgarantien wie die sicheren Sichten bereitstellen.

[0012] In einer Ausführungsform stellt eine sichere Sicht im Vergleich zu einer regulären Sicht mehrere Sicherheitsgarantien bereit. In einer Ausführungsform deckt die sichere Sicht die Sichtdefinition für die Nicht-Eigentümer der Sichten nicht auf. Dies hat Auswirkungen auf verschiedene Operationen, die auf das Datenwörterbuch zugreifen. In einer Ausführungsform deckt die sichere Sicht die Informationen über irgendwelche der Sicht zugrundeliegenden Daten, einschließlich der durch die Sicht verarbeiteten Datenmenge, der Tabellen, auf die durch die Sicht zugegriffen wird, usw., nicht auf. Dies beeinflusst die Statistiken, die über die Anzahl der in einer Abfrage abgetasteten Bytes und Partitionen angezeigt werden, und was im Abfrageprofil für eine Abfrage angezeigt wird, die sich auf eine sichere Sicht bezieht. In einer Ausführungsform deckt die sichere Sicht keine Daten aus Tabellen auf, auf die durch die Sicht zugegriffen wird, die durch die Sicht herausgefiltert werden. In einer derartigen Ausführungsform kann ein Klientenkonto, das einer nicht sicheren Sicht zugeordnet ist, durch das Ausnutzen von Abfrageoptimierungen, die verursachen können, dass die Anwenderausdrücke vor den Sicherheitsausdrücken (z. B. den Filtern und den Zusammenführungen) ausgewertet werden, auf die Daten, die herausgefiltert würden, zugreifen. In einer derartigen Ausführungsform kann, um diese Anforderung zu erreichen, der Satz von Abfrageoptimierungen, der auf eine Abfrage angewendet werden kann, die eine sichere Sicht enthält, eingeschränkt werden, um zu garantieren, dass die Anwenderausdrücke, die Daten verraten können, nicht ausgewertet werden, bevor die Sicht gefiltert ist.

[0013] Der Anmelder erkennt, dass es vorteilhaft sein kann, ähnliche Informationslecks zu verhindern. Der Anmelder stellt hier durch das Einführen einer Syntax und Unterstützung für sichere Sichten Verfahren, Systeme und Vorrichtungen zum Verhindern von Informationslecks dar. Eine derartige Syntax und Unterstützung für sichere Sichten kann verhindern, dass irgendwelche der Daten der Basistabelle innerhalb der Sicht einem Anwenderkonto verraten werden, und kann die Unterstützung zum Ändern vorhandener Sichten durch das Setzen eines „Sicher"-Merkers in der vorhandenen Sicht erweitern. Der Sicher-Merker kann angeben, dass die zuvor unsichere Sicht nun als eine sichere Sicht etikettiert ist. Mit der Einführung sicherer Sichten kann ein Anwender explizit über die Sicherheit der Sichtdefinitionen argumentieren und dieses Niveau der Sicherheit anderen Konten aufdecken, so dass der Anwender einen speziellen Anwendungsfall der Sichten implementieren kann.

[0014] Mandantenfähige Datenbanken oder mandantenfähige Datenlager unterstützen mehrere verschiedene Kundenkonten auf einmal. Als ein Beispiel ist **Fig. 1** ein schematischer Blockschaltplan, der eine mandantenfähige Datenbank oder ein mandantenfähiges Datenlager veranschaulicht, die bzw. das viele verschiedene Kundenkonten **A1**, **A2**, **A3**, **An** usw. unterstützt. Die Kundenkonten können durch mehrere Sicherheitssteuerungen getrennt sein, einschließlich unterschiedlicher einheitlicher Ressourcenzeiger (URLs), mit denen eine Verbindung hergestellt werden kann, unterschiedlicher Zugangsberechtigungsnachweise, unterschiedlicher Datenspeicherorte (wie z. B. die **S3** Buckets der Amazon-Webdienste) und unterschiedlicher Verschlüsselungsschlüssel auf Kontenebene. Folglich kann es jedem Kunden nur erlaubt sein, die eigenen Daten des

Kunden zu sehen, zu lesen und/oder zu schreiben. Durch den Entwurf kann es für einen Kunden unmöglich sein, die Daten eines weiteren Kunden zu sehen, zu lesen oder zu schreiben. In einigen Fällen ist die strikte Trennung der Kundenkonten das Rückgrat eines mandantenfähigen Datenlagers oder Datenbanksystems.

[0015] Der Anmelder erkennt, dass es manchmal wünschenswert sein kann, das kontenübergreifende gemeinsame Benutzen von Daten zu erlauben. Kein aktuelles mandantenfähiges Datenbanksystem erlaubt jedoch das gemeinsame Benutzen von Daten zwischen verschiedenen Kundenkonten in einer sofortigen und leicht steuerbaren Weise ohne Kopieren. Vorhandene Online-Datenlager, wie z. B. Amazon Redshift0, unterstützen das gemeinsame Benutzen von Daten ohne das physische Kopieren von Daten nicht. Vor-Ort-Datenbanksysteme, wie z. B. der Oracle Database Server® oder der Microsoft SQL Server0, unterstützen das gemeinsame Online-Benutzen von Daten infolge der physischen Verteilung einzelner Datenbankinstallationen definitionsgemäß nicht.

[0016] Basierend auf dem Vorhergehenden hat der Anmelder Systeme, Verfahren und Vorrichtungen für ein mandantenfähigen Online-Datenbanksystem entwickelt. Einige Ausführungsformen ermöglichen die Implementierung sicherer Sichten für das gemeinsame Benutzen von Daten ohne Kopieren zwischen verschiedenen Kundenkonten und können die Daten ohne die Notwendigkeit, die Daten zu kopieren, sofort zugänglich machen. Einige Ausführungsformen stellen den Zugriff auf die Daten unter Verwendung feinkörniger Steuerungen bereit, um die Trennung der gewünschten Daten beizubehalten, während der Zugriff auf die Daten ermöglicht wird, die ein Kunde gemeinsam benutzen möchte.

[0017] Gemäß einer Ausführungsform enthält ein computerimplementiertes Verfahren das Erzeugen eines Mitbenutzungsobjekts in einem ersten Konto, das eine Mitbenutzungsrolle enthält. Das Verfahren enthält das Zuordnen von Sichtprivilegien dem Mitbenutzungsobjekt, so dass eine zugrundeliegende Einzelheit des Mitbenutzungsobjekts eine Definition einer sicheren Sicht umfasst. Das Verfahren enthält das Bewilligen kontenübergreifender Zugriffsrechte auf die Mitbenutzungsrolle oder das Mitbenutzungsobjekt im ersten Konto einem zweiten Konto. Das Verfahren enthält ferner das Empfangen einer Anforderung vom zweiten Konto, um auf die Daten oder die Dienste des ersten Kontos zuzugreifen, und das Bereitstellen einer Antwort für das zweite Konto basierend auf den Daten oder Diensten des ersten Kontos. Das Verfahren ist so, dass die zugrundeliegende Einzelheit des Mitbenutzungsobjekts, das die Definition einer sicheren Sicht umfasst, vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist.

[0018] Gemäß einer Ausführungsform enthält ein mandantenfähiges Datenbanksystem eine kontoübergreifende Rechtekomponente, eine Aliaskomponente, eine Zugriffskomponente und eine Komponente für sichere Sichten. Die kontenübergreifende Rechtekomponente ist konfiguriert, zu bestimmen, dass kontenübergreifende Zugriffsrechte auf ein oder mehrere gemeinsam benutzte Betriebsmittel einem Zielkonto durch ein Mitbenutzerkonto bewilligt worden sind. Die Aliaskomponente ist konfiguriert, im Zielkonto ein Aliasobjekt zu erzeugen, das einem gemeinsam benutzten Betriebsmittel des einen oder der mehreren gemeinsam benutzten Betriebsmittel im Mitbenutzerkonto entspricht. Die Zugriffskomponente ist konfiguriert, dem Zielkonto zu ermöglichen, unter Verwendung einer an das Aliasobjekt gerichteten Anforderung auf das gemeinsam genutzte Betriebsmittel zuzugreifen. Die Komponente für sichere Sichten ist konfiguriert, eine sichere Sicht für ein Mitbenutzungsobjekt, ein Datenfeld eines Mitbenutzungsobjekts, ein Datenfeld eines Datenbankobjekts usw. zu definieren.

[0019] Gemäß einer weiteren Ausführungsform enthält ein Verfahren zum Bereitstellen von Zugriff auf gemeinsam benutzte Daten in einer mandantenfähigen Datenbank das Bestimmen, dass ein erstes Konto einem zweiten Konto Zugriff auf ein Datenbankobjekt des ersten Kontos bewilligt hat. Das Verfahren enthält das Implementieren von Privilegien für das Datenbankobjekt des ersten Kontos, so dass ein Datenfeld des Datenbankobjekts eine Definition einer sicheren Sicht umfasst. Das Verfahren enthält das Empfangen einer Anforderung vom zweiten Konto, auf das Datenobjekt zuzugreifen, das dem ersten Konto entspricht. Das Verfahren enthält das Verarbeiten der Anforderung unter Verwendung eines virtuellen Lagers, das dem ersten Konto entspricht. Das virtuelle Lager enthält einen Satz von einem oder mehreren Rechenknoten, die konfiguriert sind, auf Daten in einer Speicherschicht zugreifen, die dem ersten Konto entspricht, um eine Antwort auf die Anforderung zu erzeugen. Das Verfahren ist so, dass das Datenfeld des Datenbankobjekts, das die Definition einer sicheren Sicht umfasst, vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist.

[0020] Gemäß einer Ausführungsform deckt ein System für Abfragen, die sichere Sichten enthalten, weder die Menge der abgetasteten Daten (entweder in Form von Bytes oder in Form von Partitionen) noch die Gesamtmenge der Daten auf. Dies kann die Informationen vor Konten schützen, die nur Zugriff auf eine Teilmenge der Daten haben. Ein Konto kann jedoch basierend auf den Leistungseigenschaften der Abfrage geschätzte Beobachtungen über die Menge der zugrundeliegenden Daten machen. Eine Abfrage, die doppelt so lange läuft, kann z. B. doppelt so viele Daten verarbeiten. Während eine derartige Beobachtung bestenfalls approximativ ist, kann es in eini-

gen Fällen unerwünscht sein, dass sogar diese Ebene der Informationen aufgedeckt wird. In derartigen Fällen sind die Systeme, Verfahren und Vorrichtungen der Offenbarung konfiguriert, die Daten pro Anwender/Rolle zu materialisieren, anstatt die Sichten auf die Basisdaten einem Konto aufzudecken. Im Fall einer Widget-Tabelle, die Daten über mehrere variierende Widgets umfasst, kann für jede Rolle, die Zugriff auf eine Teilmenge der Widget-Tabelle hat, eine Tabelle erzeugt werden, wobei diese Tabelle nur die Teilmenge der Widgets enthält, die durch diese Rolle zugänglich sind, wobei dieser Rolle der Zugriff nur auf ihre Tabelle bewilligt würde. Eine derartige Implementierung kann im hohen Grade sichere Sichten bereitstellen und kann in bestimmten Fällen vorteilhaft sein.

[0021] Die hier offenbarten Ausführungsformen können in skalaren und tabellenwertigen anwenderdefinierten Funktionen (UDFs) unterstützt sein. Es können verschiedene Sprachen, einschließlich SQL und Javascript, für jede der skalaren und tabellenwertigen UDFs verwendet werden.

[0022] In einer Ausführungsform enthält eine skalare SQL-UDF einen SQL-Ausdruck, der Parameter enthält, die durch Argumente ersetzt werden können, die im Funktionsaufruf erscheinen. Der Ausdruck kann einen Abfrageausdruck (d. h., eine Unterabfrage) enthalten, wobei in einer Ausführungsform garantiert sein muss, dass der Abfrageausdruck eine einzelne Zeile und eine einzelne Spalte zurückgibt. Zur Kompilierungszeit wird ein Funktionsaufruf durch den SQL-Ausdruck ersetzt, wobei die Parameter im Ausdruck durch die Argumente des Aufrufs ersetzt werden. Das Ergebnis kann ein einfacher SQL-Ausdruck oder eine Unterabfrage sein (kann korreliert oder nicht korreliert sein). Weil eine skalare SQL-UDF beliebige SQL-Ausdrücke enthalten kann, kann in einer Ausführungsform die sichere UDF konfiguriert sein, (1) die Sichtdefinition für die Nicht-Eigentümer der Sicht nicht aufzudecken und (2) keine Informationen über irgendwelche zugrundeliegende Daten der Sicht aufzudecken.

[0023] In einer Ausführungsform ist eine tabellenwertige SQL-UDF zu einer Sicht ähnlich, wobei die Abfrage Parameter enthalten kann. Im Ergebnis können tabellenwertige UDFs als korrelierte Tabellen erscheinen und während der Abfragekompilierung dekorreliert werden. Weil eine tabellenwertige SQL-UDF beliebige SQL-Ausdrücke enthalten kann, können in einer Ausführungsform die sicheren UDFs konfiguriert sein, (1) die Sichtdefinition für die Nicht-Eigentümer der Sicht nicht aufzudecken, (2) die Informationen über irgendwelche zugrundeliegenden Daten der Sicht nicht aufzudecken und (3) die Daten aus den Tabellen, auf die durch die Sicht zugegriffen wird und die durch die Sicht herausgefiltert werden, nicht aufzudecken.

[0024] In einer Ausführungsform wird eine skalare Javascript-UDF in eine eingebaute Funktion umgesetzt, die ein Teil des Javascript-Codes für jede Zeile auswertet, die verarbeitet wird, und die Argumente aus diesen Zeilen in die Funktion weiterleitet. Weil eine Javascript-UDF nicht auf die Datenbanktabellen zugreifen kann, kann die UDF in einer Ausführungsform konfiguriert sein, (1) die Informationen über irgendwelche zugrundeliegende Daten der Sicht nicht aufzudecken und (2) die Daten aus den Tabellen, auf die durch die Sicht zugegriffen wird und die durch die Sicht herausgefiltert werden, nicht aufzudecken.

[0025] In einer Ausführungsform wird eine tabellenwertige Javascript-UDF verarbeitet, so dass die Zeilen gemeinsam gruppiert werden und der Javascript-Code durch das Weitergeben der Ausdrücke von den Zeilen in die Funktion pro verarbeitete Gruppe und verarbeitete Zeile ausgewertet wird. Weil eine Javascript-URDF nicht auf die Datenbanktabellen zugreifen kann, kann die UDF in einer Ausführungsform konfiguriert sein, (1) die Informationen über irgendwelche zugrundeliegende Daten der Sicht nicht aufzudecken und (2) die Daten aus den Tabellen, auf die durch die Sicht zugegriffen wird und die durch die Sicht herausgefiltert werden, nicht aufzudecken.

[0026] Die hier offenbarten Ausführungsformen stellen eine Technik bereit, die für das gemeinsame Benutzen von Daten zwischen verschiedenen Kundenkonten in einer Online-Datenbank und/oder einem Online-Datenlager verwendet werden kann. In einer Ausführungsform wird das gemeinsame Benutzen von Daten unter Verwendung kontoübergreifender Rollenbewilligungen ermöglicht, die es den Anwendern in einem Kundekonto ermöglichen, in einer gesteuerten und sofortigen Weise ohne die Notwendigkeit, Daten zu kopieren, auf die Daten in einem weiteren Kundenkonto zuzugreifen. Die gemeinsam benutzten Daten können z. B. innerhalb des gemeinsam benutzten Kontos verbleiben, während einem fremden Konto erlaubt wird, innerhalb seiner Rechte (z. B. Lesen, Schreiben usw.) Abfragen oder andere Prozesse in dem Konto auszuführen.

[0027] Die hier offenbarten Ausführungsformen können für verschiedene Anwendungen verwendet werden. In einer Ausführungsform kann ein Datenbankeigentümer einen Musterdatensatz mit neuen Kundenkonten für Tutorials und/oder eine Einführung gemeinsam benutzen. In einer Ausführungsform können Demodatensätze von einem Konto aus mit Demokonten gemeinsam benutzt werden, um ein Datenlager oder einen anderen Dienst für potentielle Kunden zu demonstrieren. In einer Ausführungsform können die durch einen Datenbankanbieter oder Host gesammelten Benutzungsdaten leicht mit Kundenkonten gemeinsam benutzt werden. In einer Ausführungsform können öffentlich zugängliche Datensätze von einem zentralen Konto mit den Kundenkon-

ten gemeinsam benutzt werden, so dass die Kundenkonten die Daten nicht selbst aufnehmen müssen. In einer Ausführungsform können die Kunden die Daten mit anderen Kunden für einen leichteren Zugriff auf die gemeinsam benutzten Betriebsmittel ohne die Notwendigkeit, die Daten physisch zu kopieren oder zu übertragen, gemeinsam benutzen. In einer Ausführungsform können die Kunden die Daten leicht mit einem Manager zur Untersuchung von Unterstützungsproblemen gemeinsam benutzen. Gemäß einer Ausführungsform können die Kunden oder andere Kontoinhaber/-eigentümer die Informationen über ihre Daten auf einem Marktplatz zum Kauf/Abonnement auflisten. Einige Ausführungsformen stellen z. B. einen Marktplatz bereit, um Datensätze zwischen den Kundenkonten einer mandantenfähigen Datenbank oder eines mandantenfähigen Datenlagers zu kaufen und zu verkaufen.

[0028] Im Folgenden wird eine ausführliche Beschreibung der Systeme und Verfahren, die mit den Ausführungsformen der vorliegenden Offenbarung konsistent sind, bereitgestellt. Während mehrere Ausführungsformen beschrieben werden, sollte erkannt werden, dass diese Offenbarung nicht auf irgendeine Ausführungsform eingeschränkt ist, sondern stattdessen zahlreiche Alternativen, Modifikationen und Äquivalente umfasst. Während in der folgenden Beschreibung zahlreiche spezifische Einzelheiten dargelegt werden, um ein umfassendes Verständnis der hier offenbarten Ausführungsformen bereitzustellen, können einige Ausführungsformen ohne einige oder alle dieser Einzelheiten praktiziert werden. Überdies wurde für den Zweck der Klarheit bestimmtes technisches Material, das im Stand der Technik bekannt ist, nicht ausführlich beschrieben, um ein unnötiges Verbergen der Offenbarung zu vermeiden.

[0029] Bezüglich der Figuren ist nun **Fig. 1** ein schematischer Blockschaltplan, der eine mandantenfähigen Datenbank oder ein mandantenfähigen Datenlager veranschaulicht, die bzw. das viele verschiedene Kundenkonten **A1**, **A2**, **A3**, **An** usw. unterstützt. Die Kundenkonten können durch mehrere Sicherheitssteuerungen getrennt sein, einschließlich unterschiedlicher einheitlicher Ressourcenzeiger (URLs), mit denen eine Verbindung hergestellt werden kann, unterschiedlicher Zugangsberechtigungsnachweise, unterschiedlicher Datenspeicherorte (wie z. B. die **S3** Buckets der Amazon-Webdienste) und unterschiedlicher Verschlüsselungsschlüssel auf Kontenebene. Folglich kann es jedem Kunden nur erlaubt sein, die eigenen Daten des Kunden zu sehen, zu lesen und/oder zu schreiben. Durch den Entwurf kann es für einen Kunden unmöglich sein, die Daten eines weiteren Kunden zu sehen, zu lesen oder zu schreiben. In einigen Fällen ist die strikte Trennung der Kundenkonten das Rückgrat eines mandantenfähigen Datenlagers oder Datenbanksystems.

[0030] **Fig. 2** ist eine schematische graphische Darstellung eines Systems **200** zum Bereitstellen von und Zugreifen auf Datenbankdaten oder -dienste. Das System **200** enthält ein Datenbanksystem **202**, einen oder mehrere Server **204** und ein Client-Rechensystem **206**. Das Datenbanksystem **202**, der eine oder die mehreren Server **204** und/oder das Client-Rechensystem **206** können über ein Netz **208**, wie z. B. das Internet, miteinander kommunizieren. Der eine oder die mehreren Server **204** und/oder das Client-Rechensystem **206** können z. B. über das Netz **208** auf das Datenbanksystem **202** zugreifen, um eine Datenbank abzufragen und/oder Daten von einer Datenbank zu empfangen. Die Daten von der Datenbank können durch den einen oder die mehreren Server **204** oder das Client-Rechensystem **206** für irgendeinen Typ von Rechenanwendung verwendet werden. In einer Ausführungsform ist das Datenbanksystem **202** ein mandantenfähiges Datenbanksystem, das Daten für mehrere verschiedene Konten hostet.

[0031] Das Datenbanksystem **202** enthält eine Mitbenutzungskomponente **210** und einen Speicher **212**. Der Speicher **212** kann Speichermedien zum Speichern von Daten enthalten. Der Speicher **212** kann z. B. eine oder mehrere Speichervorrichtungen zum Speichern von Datenbanktabellen, Schemata, Verschlüsselungsschlüsseln, Datendateien oder irgendwelchen anderen Daten enthalten. Die Mitbenutzungskomponente **210** kann Hardware und/oder Software zum Ermöglichen des kontenübergreifenden gemeinsamen Benutzens von Daten oder Diensten und/oder zum Zuordnen von Sichtprivilegien zu Daten oder Diensten enthalten. Die Mitbenutzungskomponente **210** kann z. B. von entfernten Vorrichtungen empfangene Abfragen/Anweisungen verarbeiten, um auf die gemeinsam benutzten Daten zuzugreifen oder die Daten gemeinsam zu benutzen. Die Abfragen/Anweisungen können von den Servern **204** oder dem Client-Rechensystem **206** empfangen werden. In einer Ausführungsform ist die Mitbenutzungskomponente **210** konfiguriert, das gemeinsame Benutzen von Daten zwischen den Konten zu ermöglichen, ohne doppelte Kopien von Tabellen, Daten oder dergleichen außerhalb des Mitbenutzungskontos zu erzeugen. Die Mitbenutzungskomponente kann z. B. ermöglichen, dass die einem Mitbenutzungskonto zugewiesenen Computerbetriebsmittel irgendwelche Abfragen oder Anweisungen, die durch ein fremdes Konto bereitgestellt werden, ausführen.

[0032] In einer Ausführungsform empfängt die Mitbenutzungskomponente **210** von einem Client-Rechensystem **206** eine Angabe, dass eine zugrundeliegende Einzelheit eines Mitbenutzungsobjekts einer Definition einer sicheren Sicht zugeordnet werden sollte. Der Client kann eine sichere Sicht unter Verwendung des SICHER-Schlüsselworts in SICHT ER-

ZEUGEN definieren, wobei der Client die SICHER-Eigenschaft in einer Sicht unter Verwendung eines SICHT-ÄNDERN-Befehls setzen oder untergehen lassen kann. In einer Ausführungsform ist die Mitbenutzungskomponente **210** konfiguriert, jeden der Client-Befehle auszuführen, um die sichere Sicht zu definieren, während in einer alternativen Ausführungsform die Mitbenutzungskomponente **210** konfiguriert ist, z. B. basierend auf früheren Einstellungen, die vom Client-Rechensystem **206** empfangen worden sind, automatisch eine sichere Sicht zu definieren. Um einer zugrundeliegenden Einzelheit eines Mitbenutzungsobjekts Sichtprivilegien zuzuordnen, ändert die Mitbenutzungskomponente **210** den Parser, so dass er das „sichere“ Schlüsselwort vor dem Sichtnamen und der neuen Regel zum Ändern der Sicht unterstützt. Die Sicht ist effektiv als eine Tabelle gespeichert, wobei die Änderung das Ändern des Tabellendaten-Beständigkeitsobjekts (DPO) und das Hinzufügen eines „Sicher“-Merkers beinhaltet, um (zusätzlich zu dem Sichttext, der nun das „Sicher“-Etikett enthält) anzugeben, dass die Sicht eine sichere Sicht ist.

[0033] In einer Ausführungsform stellt die Mitbenutzungskomponente **210** sicher, dass die Definition einer sicheren Sicht vor den Anwendern oder Klientenkonten verborgen ist, die nicht der Eigentümer der Sicht (oder der zugrundeliegenden Daten) sind. Die Mitbenutzungskomponente **210** kann durch das Ändern der Ausgabe von Befehlen, so dass die Ergebnisse wie üblich an den Eigentümer einer Sicht zurückgegeben werden, aber die Definition einer sicheren Sicht nicht an einen Nicht-Eigentümer, der Zugriff auf die Sicht hat, zurückgegeben wird, sicherstellen, dass die Definition einer sicheren Sicht verborgen ist.

[0034] In einer Ausführungsform können die Speicher- und Rechenbetriebsmittel für ein Datenbanksystem **100** logisch und/oder physisch getrennt sein. In einer Ausführungsform ist der Speicher ein gemeinsames, gemeinsam benutztes Betriebsmittel für alle Konten. Die Rechenbetriebsmittel können unabhängig pro Konto als virtuelle Lager aufgebaut sein. In einer Ausführungsform ist ein virtuelles Lager ein Satz von Rechenknoten, die auf die Daten in einer Speicherschicht zugreifen und ein Abfrageergebnis berechnen. Das Trennen der Rechenknoten oder der Betriebsmittel vom Speicher ermöglicht das unabhängige Skalieren jeder Schicht. Die Trennung des Speichers und des Rechnens ermöglicht außerdem, dass gemeinsam benutzte Daten durch verschiedene Konten unabhängig verarbeitet werden können, ohne dass die Berechnung in einem Konto die Berechnung in anderen Konten beeinflusst. Das heißt, in wenigstens einigen Ausführungsformen gibt es keine Konkurrenz zwischen den Rechenbetriebsmitteln, wenn Abfragen auf gemeinsam benutzte Daten ausgeführt werden.

[0035] **Fig. 3** ist ein schematischer Blockschaltplan einer mandantenfähigen Datenbank **300**, der die Trennung der Speicher- und Rechenbetriebsmittel veranschaulicht. Die mandantenfähige Datenbank **300** kann z. B. ein Datenlager sein, in dem mehrere verschiedener Konten (**A1**, **A2**, **A3** bis **An**) gehostet sind. In **Fig. 3** weist das Konto **A1** drei laufende virtuelle Lager auf, weist das Konto **A2** ein laufendes virtuelles Lager auf und weist das Konto **A3** kein laufendes virtuelles Lager auf. In einer Ausführungsform haben alle diese virtuellen Lager Zugriff auf die Speicherschicht, die von den Rechenknoten der virtuellen Lager getrennt ist. In einer Ausführungsform können die virtuellen Lager basierend auf einer aktuellen Arbeitslast für ein Konto dynamisch bereitgestellt oder entfernt werden.

[0036] In einer Ausführungsform verwendet ein Datenbanksystem **300** Objekthierarchien in den Konten. Jedes Kundenkonto kann z. B. Objekthierarchien enthalten. Die Objekthierarchien sind oft in den Datenbanken verwurzelt. Die Datenbanken können z. B. Schemata enthalten, wobei die Schemata wiederum Objekte, wie z. B. Tabellen, Sichten, Folgen, Dateiformate und Funktionen, enthalten können. Jedes dieser Objekte dient einem speziellen Zweck: die Tabellen speichern relationale oder halbstrukturierte Daten; die Sichten definieren logische Abstraktionen über die gespeicherten Daten; die Folgen stellen Mittel zur Erzeugung ständig zunehmender Zahlen bereit; die Dateiformate definieren Wege, um aufgenommene Datendateien zu parsen; und die Funktionen enthalten anwenderdefinierte Ausführungsprozeduren. In den Ausführungsformen, wie sie hier offenbart werden, können die Sichten den Definitionen sicherer Sichten zugeordnet sein, so dass die zugrundeliegenden Daten, die der Sicht zugeordnet sind, für die Konten der Nicht-Eigentümer, die Zugriff auf die Sicht haben, verborgen sind.

[0037] **Fig. 4** ist ein schematischer Blockschaltplan, der Objekthierarchien innerhalb der Kundenkonten veranschaulicht. Spezifisch können die Konten Hierarchien von Objekten enthalten, auf die in einer Datenbank verwiesen werden kann. Das Kundenkonto **A1** enthält z. B. zwei Datenbankobjekte **D1** und **D2**. Das Datenbankobjekt **D1** enthält ein Schemenobjekt **S1**, das wiederum ein Tabellenobjekt **T1** und ein Sichtobjekt **V1** enthält. Das Datenbankobjekt **D2** enthält ein Schemenobjekt **S2**, das ein Funktionsobjekt **F2**, ein Folgenobjekt **Q2** und ein Tabellenobjekt **T2** enthält. Das Kundenkonto **A2** enthält ein Datenbankobjekt **D3** mit einem Schemenobjekt **S3** und einem Tabellenobjekt **T3**. Die Objekthierarchien können steuern, wie auf die Objekte, die Daten, die Funktionen oder die anderen Informationen oder Dienste eines Konto- oder Datenbanksystems zugegriffen oder verwiesen wird.

[0038] In einer Ausführungsform implementiert ein Datenbanksystem eine rollenbasierte Zugriffssteuerung, um den Zugriff auf die Objekte in den Kundenkonten zu steuern. Im Allgemeinen besteht die rollenbasierte Zugriffssteuerung aus zwei Grundprinzipien: Rollen und Bewilligungen. In einer Ausführungsform sind die Rollen spezielle Objekte in einem Kundenkonto, die den Anwendern zugewiesen sind. Die Bewilligungen zwischen den Rollen und den Datenbankobjekten definieren, welche Privilegien eine Rolle bei diesen Objekten aufweist. Eine Rolle, die eine Benutzungsbewilligung für eine Datenbank aufweist, kann z. B. diese Datenbank „sehen“, wenn sie den Befehl „Zeige Datenbanken“ ausführt; eine Rolle, die eine Auswahlbewilligung für eine Tabelle aufweist, kann aus dieser Tabelle lesen, aber nicht in die Tabelle schreiben. Die Rolle würde eine Modifikationsbewilligung für die Tabelle aufweisen müssen, um in sie schreiben zu können.

[0039] **Fig. 5** ist ein schematischer Blockschaltplan, der den rollenbasierten Zugriff auf Objekte in Kundenkonten veranschaulicht. Ein Kundenkonto **A1** enthält eine Rolle **R1**, die Bewilligungen für alle Objekte in der Objekthierarchie aufweist. Angenommen, dass diese Bewilligungen Benutzungsbewilligungen zwischen **R1** und **D1**, **D2**, **S1**, **S2** und Auswahlbewilligungen zwischen **R1** und **T1**, **V1**, **F2**, **Q2**, **T2** sind, kann ein Anwender mit der aktivierten Rolle **R1** alle Objekte sehen und Daten aus allen Tabellen, Sichten und Folgen lesen und die Funktion **F2** innerhalb des Kontos **A1** ausführen. Das Kundenkonto **A2** enthält eine Rolle **R3**, die Bewilligungen für alle Objekte in der Objekthierarchie aufweist. Angenommen, dass diesen Bewilligungen Benutzungsbewilligungen zwischen **R3** und **D3**, **S3** sind und eine Bewilligung zwischen **R3** und **T3** auswählen, kann ein Anwender mit der aktivierten Rolle **R3** alle Objekte sehen und Daten aus allen Tabellen, Sichten und Folgen innerhalb des Kontos **A2** lesen.

[0040] **Fig. 6** veranschaulicht eine Benutzungsbewilligung zwischen Rollen. Bei der rollenbasierten Zugriffssteuerung ist es außerdem möglich, die Benutzung von einer Rolle zu einer weiteren Rolle zu bewilligen. Eine Rolle, die eine Benutzungsbewilligung für eine weitere Rolle aufweist, „übernimmt“ alle Zugriffsprivilegien der anderen Rolle. Die Rolle **R2** weist z. B. eine Benutzungsbewilligung für die Rolle **R1** auf. Ein Anwender (z. B. mit entsprechenden Autorisierungseinzelheiten) mit der aktivierten Rolle **R2** kann alle Objekte sehen und aus allen Objekten lesen, weil die Rolle **R2** alle Bewilligungen von der Rolle **R1** übernimmt.

[0041] Gemäß einer Ausführungsform werden Benutzungsbewilligungen über verschiedene Konten bewilligt. Ein Konto, das Daten gemeinsam benutzt, kann hier als ein „Mitbenutzerkonto“ bezeichnet werden, während ein Konto, mit dem die Daten gemeinsam benutzt werden, hier als ein „Zielkonto“ bezeichnet werden kann. In einer Ausführungsform ermöglichen die hier offenbarten Ausführungsformen eine sofortige, leicht steuerbare kontenübergreifende gemeinsame Benutzung von Daten ohne Kopieren, die die Definitionen sicherer Sichten implementieren kann, so dass das Zielkonto die zugrundeliegenden Einzelheiten oder Sichtdefinitionen der Sicht nicht sehen kann. In einer Ausführungsform kann ein Mitbenutzerkonto ein Mitbenutzungsobjekt erzeugen, um die Daten mit einem weiteren Konto gemeinsam zu benutzen. Innerhalb des Mitbenutzungsobjekts kann eine Rolle erzeugt werden, wobei ein Anwender des Mitbenutzerkontos angeben kann, dass Zugriffsrechte oder -bewilligungen für die Rolle und/oder für fremde (oder Zielkonten) verfügbar sind, denen gemäß der Rolle Rechte bewilligt werden. Ein Sichtprivileg kann einer zugrundeliegenden Einzelheit des Mitbenutzungsobjekts, wie z. B. einem Datenfeld einer zugrundeliegenden Datenbanktabelle, einer Datenmenge innerhalb der Datenbanktabelle und/oder einem Strukturelement einer zugrundeliegenden Datenbanktabelle des Mitbenutzungsobjekts, zugeordnet sein. Das Sichtprivileg kann eine Definition einer sicheren Sicht angeben, die angibt, dass das Zielkonto keinen Zugriff auf die zugrundeliegende Einzelheit des Mitbenutzungsobjekts hat. Ein Zielkonto kann dann in der Lage sein, die Mitbenutzungsobjekte oder -rollen in einem anderen Konto zu identifizieren, für das dem Zielkonto Rechte oder Zugriff bewilligt worden sind. In einer Ausführungsform können die Mitbenutzungsobjekte in einem Mitbenutzerkonto unter Verwendung von Aliasobjekten und kontoübergreifenden Rollenbewilligungen in das Zielkonto importiert werden. In einer Ausführungsform können derartige Mitbenutzungsobjekte als eine sichere Sicht in das Zielkonto importiert werden, so dass eine oder mehrere zugrundeliegende Einzelheiten des Mitbenutzungsobjekts vor dem Zielkonto verborgen sind.

[0042] In einer Ausführungsform wird das gemeinsame Benutzen von Daten durch das Konto eingeleitet, das die gemeinsam zu benutzenden Daten besitzt. Das Mitbenutzerkonto erzeugt z. B. einen neuen Objekttyp, das Mitbenutzungsobjekt. Das Mitbenutzungsobjekt weist einen eindeutigen Namen auf, um innerhalb des Mitbenutzerkontos identifiziert zu werden. Der Name muss z. B. vielleicht innerhalb eines Kontos, aber nicht notwendigerweise kontoübergreifend eindeutig sein. Die Mitbenutzungsobjekte können erzeugt, modifiziert und gelöscht werden, indem auf sie über ihren Namen im Mitbenutzerkonto verwiesen wird.

[0043] In einer Ausführungsform enthält jedes Mitbenutzungsobjekt eine einzige Rolle. Bewilligungen zwischen dieser Rolle und den Objekten definieren, welche Objekte gemeinsam benutzt werden und mit welchen Privilegien diese Objekte gemeinsam be-

nutzt werden. Die Rolle und die Bewilligungen können bei der Implementierung einer rollenbasierten Zugriffssteuerung zu irgendeinem anderen Rollen- und Bewilligungssystem ähnlich sein. Durch das Modifizieren des Satzes von Bewilligungen, die der Rolle in einem Mitbenutzungsobjekt beigefügt sind, können (durch das Hinzufügen von Bewilligungen zu der Rolle) mehr Objekte gemeinsam benutzt werden, können (durch das Aufheben von Bewilligungen der Rolle) weniger Objekte gemeinsam benutzt werden oder können (durch das Ändern des Typs der Bewilligung, z. B. um Schreibzugriff auf ein gemeinsam benutztes Tabellenobjekt zu ermöglichen, das zuvor schreibgeschützt war) Objekte mit unterschiedlichen Privilegien gemeinsam benutzt werden.

[0044] In einer Ausführungsform enthält ein Mitbenutzungsobjekt außerdem eine Liste von Verweisen auf andere Kundenkonten. Nur diesen Konten, die sich spezifisch im Mitbenutzungsobjekt befinden, kann erlaubt sein, in diesem Mitbenutzungsobjekt nachzuschlagen, auf dieses Mitbenutzungsobjekt zuzugreifen und/oder aus diesem Mitbenutzungsobjekt zu importieren. Durch das Modifizieren der Liste der Verweise auf andere Kundenkonten kann das Mitbenutzungsobjekt für mehr Konten zugänglich gemacht oder auf weniger Konten eingeschränkt werden.

[0045] In einer Ausführungsform ist die Verarbeitung sicherer Sichten durch Transformationen des Parse-Baums, z. B. Sichtverschmelzung und Prädikattransformation, implementiert. Die kanonische Implementierung würde sein, Abfrageblöcke so zu kommentieren, so dass die Abfrageblöcke als von sicheren Sichten kommend bezeichnet werden. In einem derartigen Fall können die Abfrageblöcke nicht mit externen Abfrageblöcken (z. B. über eine Sichtverschmelzung) oder Ausdrücken (z. B. über eine Filter-Einkellerung) kombiniert werden.

[0046] In einer Ausführungsform ist die Verarbeitung sicherer Sichten durch Neuschreibvorgänge des Abfrageplanbaums während der Optimierung, z. B. während des Filter-Hochziehens und der Filter-Einkellerung, implementiert. Die Neuschreibvorgänge des Abfrageplanbaums sind so implementiert, dass kein Ausdruck, der nicht von einer sicheren Sicht stammt, nach unten unter die Sichtgrenzen geschoben werden kann. Dies kann durch das Erzeugen eines neuen Typs von Projektionsoperator (z. B. eines Operators sicherer Projektionen) implementiert werden, der sich zu einer Standardprojektion völlig gleich verhält, aber, weil er keine Standardprojektion ist, irgendwelchen der Vorbedingungen der Neuschreibregel nicht entspricht. Im Ergebnis werden die relevanten Neuschreibvorgänge nicht angewendet. Das Identifizieren, welcher Typ von Projektion erzeugt werden soll, d. h., eine Standard- oder sichere Projektion, kann implementiert werden, nachdem die Sichtabfrageblöcke als von einer Definition einer sicheren Sicht kommend bezeichnet worden sind oder nicht.

[0047] In einer Ausführungsform sind die Optimierungen für die sicheren Sichten implementiert, um einen potentiellen Leistungsverlust abzuschwächen, der historisch der Verwendung sicherer Sichten zugeordnet ist. Derartige Optimierungen werden implementiert, indem bestimmte Transformationen insofern für „sicher“ gehalten werden, als die Transformation keinen potentiellen Grund zur Besorgnis darstellt. Die Sicherheit in einem derartigen Fall bedeutet, dass die Operationen, die transformiert werden, keine Nebenwirkungen aufweisen, die sich typischerweise auf eine anwenderdefinierte Funktion beziehen, die Operationen, die von der SQL nicht leicht als unsicher identifiziert werden können, oder Operationen, die potentiell keine Informationen über den Datenwert enthüllen können, der den Fehler verursacht hat, wenn z. B. durch null dividiert wird, ausführt. Im Ergebnis enthalten die Optimierungen das Kommentieren der Ausdrücke mit ihren Sicherheitseigenschaften und dann das Ermöglichen der Transformationen, die es ermöglichen, dass ein Ausdruck durch eine Grenze sicherer Sichten geschoben wird, falls sie für sicher gehalten werden. Der Ausdruck wird für sicher gehalten, falls bekannt ist, dass der Ausdruck keine Fehler erzeugt und keine anwenderdefinierten Funktionen enthält. Ob der Ausdruck Fehler erzeugt, wird unter Verwendung eines Rahmens bestimmt, der konfiguriert ist, abhängig von dem Typ des Ausdrucks und den Eigenschaften seiner Eingaben (d. h., ob die Eingaben Fehler erzeugen) zu bestimmen, ob ein Ausdruck zur Laufzeit einen Fehler erzeugen kann. In bestimmten Implementierungen kann es wichtig sein, sicherzustellen, dass der Ausdruck keine anwenderdefinierten Funktionen enthält, weil, selbst wenn eine anwenderdefinierte Funktion nie einen Fehler erzeugt, sie den Zustand von den Zeilen erfassen kann, die aus den Anwenderergebnissen herausgefiltert werden sollten, und diesen Zustand in den Zeilen verraten kann, die nicht herausgefiltert werden.

[0048] In einer Ausführungsform enthalten weitere Optimierungen das Optimieren einer sicheren Sicht, als ob sie eine nicht sichere Sicht wäre. Dies kann implementiert sein, falls die aktuelle Rolle Zugriff auf alle zugrundeliegenden Einzelheiten (z. B. die zugrundeliegenden Tabellen, die zugrundeliegenden Datenfelder, die zugrundeliegenden Datenbankstrukturen, die zugrundeliegenden Sichten usw.) hat, auf die innerhalb des Mitbenutzungsobjekts verwiesen wird.

[0049] Zusätzlich kann die Verwendung sicherer Sichten beim gemeinsamen Benutzen von Daten optimiert werden, indem ein Filter-Hochziehen durch sichere Projektionen ermöglicht wird. Das Filter-Hochziehen kann für Unterabfrage-Neuschreibvorgänge nützlich sein und kann ferner für die transitive Prädi-

katerzeugung nützlich sein, bei der die Filter durch eine Verknüpfung hochgezogen werden, wobei, wenn sie durch das Verknüpfen zurück heruntergeschoben werden, die Filter potentiell auf beide Seiten der Verknüpfung geschoben werden können. Das Filter-Hochziehen kann durch das Kopieren des Filters beim Hochziehen durch die sichere Projektion implementiert sein, so dass das ursprüngliche Prädikat unterhalb der sicheren Projektion bleibt, um sicherzustellen, dass alle sicherheitsbezogenen Filterungen vor irgendeiner Ausdrucksauswertung außerhalb der Sicht stattfinden. Die hochgezogene Kopie des Prädikats kann für auf das Hochziehen bezogene Optimierungen verwendet werden.

[0050] Zusätzlich kann die Verwendung der Umsetzung von der äußeren Verknüpfung zur inneren Verknüpfung unterhalb der sicheren Projektionen verwendet werden, um die Verwendung sicherer Sichten beim gemeinsamen Benutzen von Daten zu optimieren. Die Umsetzung von der äußeren Verknüpfung zur inneren Verknüpfung enthält ein Filterprädikat, das unmittelbar über einer Verknüpfung erscheint, wobei das Filterprädikat die Nullen von der mit Nullen erweiterten Seite der äußeren Verknüpfung herausfiltert. Wenn eine sichere Sicht eine äußere Verknüpfung enthält und ein Anwenderfilter die Nullen herausfiltert, dann kann der Filter nicht die ganze Strecke bis zur Verknüpfung geschoben werden, wenn er nicht ein sicher zu schiebender Filter ist. Die Optimierung enthält ferner das Ausbreiten der Tatsache, dass die Nullen herausgefiltert werden können, ohne den Filter durch die sichere Projektion zu bewegen.

[0051] **Fig. 7** ist ein schematischer Blockschaltplan, der ein Mitbenutzungsobjekt **SH1** veranschaulicht. Das Kundenkonto **A1** enthält das Mitbenutzungsobjekt **SH1**. Das Mitbenutzungsobjekt **SH1** weist einen eindeutigen Namen „**SH1**“ im Kundenkonto **A1** auf. Das Mitbenutzungsobjekt **SH1** enthält die Rolle **R4** mit Bewilligungen für die Datenbank **D2**, das Schema **S2** und die Tabelle T2. Die Bewilligungen in der Datenbank **D2** und dem Schema **S2** können Benutzungsbewilligungen sein, während die Bewilligungen in der Tabelle T2 eine Auswahlbewilligung sein kann. In diesem Fall würde die Tabelle T2 im Schema **S2** in der Datenbank **D2** schreibgeschützt gemeinsam benutzt werden. Das Mitbenutzungsobjekt **SH1** enthält eine Liste von Verweisen auf andere Kundenkonten einschließlich des Kontos **A2**.

[0052] Nachdem ein Mitbenutzungsobjekt erzeugt worden ist, kann das Mitbenutzungsobjekt durch ein im Mitbenutzungsobjekt aufgelistetes Zielkonto importiert oder von dem Zielkonto auf es verwiesen werden. Das Importieren eines Mitbenutzungsobjekts von einem Mitbenutzerkonto ist z. B. von anderen Kundenkonten möglich. Ein Zielkonto kann einen Befehl ausführen, um alle zum Importieren verfügbaren Mitbenutzungsobjekte aufzulisten. Nur wenn ein Mitbenutzungsobjekt mit Verweisen erzeugt wurde, die das Zielkonto enthalten, kann das Zielkonto das Mitbenutzungsobjekt auflisten und es anschließend importieren. In einer Ausführungsform werden die Verweise auf ein Mitbenutzungsobjekt in einem weiteren Konto immer durch den Kontonamen qualifiziert. Das Zielkonto **A2** würde z. B. auf das Mitbenutzungs-SH1 im Mitbenutzerkonto **A1** mit dem beispielhaften qualifizierten Namen „A1.SH1“ verweisen.

[0053] In einer Ausführungsform kann ein Prozess oder das Importieren eines Mitbenutzungsobjekts enthalten: Erzeugen eines Aliasobjekts im Zielkonto; Verketten des Aliasobjekts mit dem obersten gemeinsam benutzten Objekt im Mitbenutzerkonto in der Objekthierarchie; Bewilligen einer Rolle in den Benutzungsprivilegien des Zielkontos dem Aliasobjekt; und Bewilligen der Benutzungsprivilegien der Rolle des Zielkontos der im Mitbenutzungsobjekt enthaltenen Rolle.

[0054] In einer Ausführungsform erzeugt ein Zielkonto, das das Mitbenutzungsobjekt oder die Daten importiert, ein Aliasobjekt. Ein Aliasobjekt ist zu einem normalen Objekt in einem Kundenkonto ähnlich. Ein Aliasobjekt weist seinen eigenen eindeutigen Namen auf, mit dem es identifiziert wird. Ein Aliasobjekt kann mit dem obersten Objekt in jeder Objekthierarchie, die gemeinsam benutzt wird, verkettet sein. Falls mehrere Objekthierarchien gemeinsam benutzt werden, können im Zielkonto mehrere Aliasobjekte erzeugt werden. Wann immer ein Aliasobjekt verwendet wird (z. B. Lesen aus dem Aliasobjekt, Schreiben in das Aliasobjekt), wird das Aliasobjekt intern durch das normale Objekt im Mitbenutzerkonto ersetzt, mit dem es verkettet ist. In dieser Weise sind die Aliasobjekte lediglich Proxy-Objekte von normalen Objekten und keine doppelten Objekte. Wenn folglich aus einem Aliasobjekt gelesen wird oder in ein Aliasobjekt geschrieben wird, beeinflussen die Operationen das ursprüngliche Objekt, mit dem das Alias verkettet ist. Wenn ein Aliasobjekt erzeugt wird, wird es wie normale Objekte der aktivierten Rolle des Anwenders bewilligt.

[0055] Zusätzlich zum Aliasobjekt wird eine Bewilligung zwischen einer Rolle im Zielkonto und der im Mitbenutzungsobjekt enthaltenen Rolle erzeugt. Dies ist eine Benutzungsbewilligung von Rolle zu Rolle über die Kundenkonten. Die rollenbasierte Zugriffssteuerung ermöglicht nun einem Anwender im Zielkonto, auf die Objekte im Mitbenutzerkonto zuzugreifen.

[0056] **Fig. 8** ist ein schematischer Blockschaltplan, der logische Bewilligungen und Verkettungen zwischen verschiedenen Konten veranschaulicht. Im Konto **A2** wird ein Datenbank-Aliasobjekt **D5** erzeugt. Das Datenbank-Alias **D5** verweist über die Verkettung **L1** auf die Datenbank **D2**. Die Rolle **R3** weist ei-

ne Benutzungsbewilligung **G1** in der Datenbank **D5** auf. Die Rolle **R3** weist eine zweite Benutzungsbewilligung **G2** für die Rolle **R4** im Kundenkonto **A1** auf. Die Bewilligung **G2** ist eine kontenübergreifende Bewilligung zwischen den Konten **A1** und **A2**. In einer Ausführungsform ermöglicht es die rollenbasierte Zugriffssteuerung einem Anwender im Konto **A2** mit der aktivierten Rolle **R3**, auf die Daten im Konto **A1** zuzugreifen. Falls z. B. ein Anwender im Konto **A2** die Daten in der Tabelle T2 lesen will, ermöglicht die rollenbasierte Zugriffssteuerung dies, weil die Rolle **R3** eine Benutzungsbewilligung der Rolle **R4** aufweist und die Rolle **R4** wiederum eine Auswahlbewilligung in der Tabelle T2 aufweist. Zur Veranschaulichung kann ein Anwender mit der aktivierten Rolle **R3** durch das Ausführen einer Abfrage oder Auswahl, die auf „D5.S2.T2“ gerichtet ist, auf **T2** zugreifen.

[0057] Die Verwendung von Objektaliassen und kontenübergreifenden Bewilligungen von einer Rolle im Zielkonto zu einer Rolle im Mitbenutzerkonto ermöglicht es den Anwendern im Zielkonto, auf die Informationen im Mitbenutzerkonto zuzugreifen. In dieser Weise kann ein Datenbanksystem das gemeinsame Benutzen von Daten zwischen verschiedenen Kundenkonten in einer sofortigen und leicht steuerbaren Weise ohne Kopieren ermöglichen. Das gemeinsame Benutzen kann sofort stattfinden, weil die Aliasobjekte und die kontoübergreifende Bewilligungen in Millisekunden erzeugt werden können. Das gemeinsame Benutzen kann ohne Kopieren erfolgen, weil in dem Prozess keine Daten kopiert werden müssen. Es können z. B. alle Abfragen oder Auswahlen direkt an das gemeinsam benutzte Objekt im Mitbenutzerkonto ausgeführt werden, ohne ein Duplikat im Zielkonto zu erzeugen. Das gemeinsame Benutzen ist außerdem leicht zu steuern, weil es leicht zu verwendende Techniken der rollenbasierten Zugriffssteuerung verwendet. Zusätzlich gibt es in den Ausführungsformen mit getrennter Speicherung und Berechnung keine Konkurrenz zwischen den Rechenbetriebsmitteln, wenn Abfragen an gemeinsam benutzte Daten ausgeführt werden. Folglich können verschiedene virtuelle Lager in verschiedenen Kundenkonten gemeinsam benutzte Daten einzeln verarbeiten. Ein erstes virtuelles Lager für ein erstes Konto kann z. B. eine Datenbankabfrage oder -anweisung unter Verwendung der Daten verarbeiten, die durch ein Mitbenutzerkonto gemeinsam benutzt werden, während ein zweites virtuelles Lager für ein zweites Konto oder das Mitbenutzerkonto eine Datenbankabfrage oder -anweisung unter Verwendung der gemeinsam benutzten Daten des Mitbenutzerkontos verarbeiten kann.

[0058] Fig. 9 ist ein schematischer Blockschaltplan einer Mitbenutzungskomponente **210**. Die Mitbenutzungskomponente **210** enthält eine kontoübergreifende Rechtekomponente **902**, eine Aliaskomponente **904**, eine Anforderungskomponente **906**, eine Zugriffskomponente **908**, eine Verarbeitungskomponente **910** und eine Komponente **912** für sichere Sichten. Die Komponenten **902-912** sind lediglich beispielhaft angegeben und können nicht alle in allen Ausführungsformen enthalten sein. Jede der Komponenten **902-912** kann z. B. in einer separaten Vorrichtung oder einem separaten System enthalten sein oder als ein Teil einer separaten Vorrichtung oder eines separaten Systems implementiert sein.

[0059] Die kontenübergreifende Rechtekomponente **902** ist konfiguriert, Rechte oder Bewilligungen zwischen den Konten zu erzeugen und zu managen. Die kontoübergreifende Rechtekomponente **902** kann ein Mitbenutzungsobjekt in einem Mitbenutzerkonto erzeugen. Ein Anwender des Mitbenutzerkontos kann z. B. eine Eingabe bereitstellen, die angibt, dass ein oder mehrere Betriebsmittel mit einem weiteren Konto gemeinsam benutzt werden sollten. In einer Ausführungsform kann der Anwender eine Option auswählen, um ein neues Mitbenutzungsobjekt zu erzeugen, so dass die Betriebsmittel mit fremden Konten gemeinsam benutzt werden können. In Reaktion auf die Anwendereingabe kann die kontoübergreifende Rechtekomponente **902** ein Mitbenutzungsobjekt im Mitbenutzerkonto erzeugen. Das Mitbenutzungsobjekt kann eine Rolle enthalten, der Zugriffsrechte auf die Betriebsmittel zum gemeinsamen Benutzen mit einem fremden Konto bewilligt werden können. Das fremde Konto kann ein Kundenkonto oder ein anderes Konto enthalten, das vom Mitbenutzerkonto getrennt ist. Das fremde Konto kann z. B. ein weiteres Konto sein, das in einem mandantenfähigen Datenbanksystem gehostet ist.

[0060] Bei der Erzeugung können dem Mitbenutzungsobjekt Rechte an einem oder mehreren Betriebsmitteln innerhalb des Mitbenutzerkontos bewilligt werden. Die Betriebsmittel können eine Datenbank, ein Schema, eine Tabelle, eine Folge oder eine Funktion des Mitbenutzerkontos enthalten. Das Mitbenutzungsobjekt kann z. B. eine Rolle (d. h., eine Mitbenutzungsrolle) enthalten, der das Recht bewilligt ist, ein Datenspeicherobjekt, wie z. B. eine Datenbank, zu lesen, auszuwählen, abzufragen oder zu modifizieren. Dem Mitbenutzungsobjekt oder der Mitbenutzungsrolle im Mitbenutzungsobjekt können Rechte ähnlich dazu bewilligt werden, wie anderen Rollen unter Verwendung einer rollenbasierten Zugriffssteuerung Rechte bewilligt werden können. Ein Anwender kann imstande sein, auf ein Konto zuzugreifen und der Mitbenutzungsrolle Rechte zu bewilligen, so dass die Mitbenutzungsrolle auf die Betriebsmittel zugreifen kann, für die beabsichtig ist, dass sie mit fremden Konten gemeinsam benutzt werden. In einer Ausführungsform kann das Mitbenutzungsobjekt eine Liste von Objekten und eine Zugriffsebene enthalten, für die die Mitbenutzungsrolle Rechte aufweist.

[0061] Das Mitbenutzungsobjekt kann außerdem spezifischen fremden Konten verfügbar gemacht oder mit ihnen verkettet werden. Das Mitbenutzungsobjekt kann z. B. eine Liste von Konten speichern, die Rechte für die Mitbenutzungsrolle oder das Mitbenutzungsobjekt im Mitbenutzerkonto aufweisen. Ein Anwender mit dem Mitbenutzerkonto kann Konten zur Liste der Konten hinzufügen oder aus der Liste der Konten entfernen. Der Anwender kann z. B. imstande sein, die Liste zu modifizieren, um zu steuern, welche Konten auf Objekte zugreifen können, die über das Mitbenutzungsobjekt gemeinsam benutzt werden. Fremden Konten, die in dem Mitbenutzungsobjekt aufgelistet oder identifiziert sind, kann mit den einer Mitbenutzungsrolle des Mitbenutzungsobjekts bewilligten Zugriffsrechten Zugriff auf die Betriebsmittel gegeben werden. In einer Ausführungsform kann ein spezifisches Konto eine Suche ausführen, um Mitbenutzungsobjekte oder Mitbenutzerkonten zu identifizieren, die mit dem spezifischen Konto gemeinsam benutzt worden sind. Eine Liste der verfügbaren Mitbenutzungsobjekte kann durch einem Anwender des Ziel- oder spezifischen Kontos betrachtet werden.

[0062] Die Aliaskomponente **904** ist konfiguriert, ein Alias für die Daten oder ein Datenobjekt zu erzeugen, die bzw. das durch ein separates Konto gemeinsam benutzt werden bzw. wird. Das Aliasobjekt kann z. B. in einem Zielkonto ein Aliasobjekt erzeugen, das einem durch ein Mitbenutzerkonto gemeinsam benutzten gemeinsam benutzten Betriebsmittel entspricht. In einer Ausführungsform wird das Aliasobjekt in Reaktion auf ein Zielkonto erzeugt, das ein gemeinsam benutztes Betriebsmittel akzeptiert oder zum ersten Mal versucht, auf ein gemeinsam genutztes Betriebsmittel zuzugreifen. Das Aliasobjekt kann als ein Alias für ein Datenobjekt für die höchste Objekthierarchie wirken, die durch das Mitbenutzerkonto gemeinsam benutzt wird (siehe z. B. **Fig. 8**, wo **D5** ein Alias für **D2** ist). Die Aliaskomponente **904** kann auch eine Verkettung zwischen dem Aliasobjekt und einem gemeinsam benutzten Objekt erzeugen (siehe z. B. **Fig. 8**, wo L1 die Verkettung zwischen **D5** und **D2** ist). Die Verkettung kann in Form einer Kennung oder eines Namens des ursprünglichen oder „echten“ Objekts erzeugt und/oder gespeichert werden. Die Verkettung **L1** in **Fig. 8** kann z. B. eine Kennung für **D2** enthalten, die im Aliasobjekt **D5** gespeichert ist und die einen eindeutigen systemweiten Namen, wie z. B. „A1. D2“, enthält.

[0063] Die Aliaskomponente **904** kann außerdem einer Rolle im Zielkonto (das Konto, mit dem das Mitbenutzerkonto Daten oder Betriebsmittel gemeinsam benutzt hat) Zugriffsrechte auf das Aliasobjekt bewilligen (siehe z. B. G1 nach **Fig. 8**). Zusätzlich kann die Aliaskomponente **904** die Rolle im Zielkonto außerdem einer Mitbenutzungsrolle im Mitbenutzungsobjekt des Mitbenutzerkontos bewilligen (siehe z. B. G2 nach **Fig. 8**). Mit dem erzeugten Aliasobjekt, einer Verkettung zwischen dem Aliasobjekt und einem Objekt im Mitbenutzerkonto und den Bewilligungen für eine Rolle im Zielkonto kann das Zielkonto frei sein, um Abfragen, Anweisungen auszuführen oder gemeinsam benutzte Daten oder Betriebsmittel im Mitbenutzerkonto zu „sehen“.

[0064] Die Anforderungskomponente **906** ist konfiguriert, eine Anforderung von einem Konto zu empfangen, um auf ein gemeinsam benutztes Betriebsmittel in einem anderen Konto zuzugreifen. Die Anforderung kann eine Datenbankabfrage, eine Auswahlanweisung oder dergleichen enthalten, um auf ein Betriebsmittel zuzugreifen. In einer Ausführungsform enthält die Anforderung eine an ein Aliasobjekt des anfordernden Kontos gerichtete Anforderung. Die Anforderungskomponente **906** kann ein Betriebsmittel identifizieren, mit dem das Aliasobjekt verkettet ist, wie z. B. eine Datenbank oder eine Tabelle in einem Mitbenutzerkonto. Die Anforderungskomponente **906** kann das verkettete Objekt basierend auf einer Kennung des Aliasobjekts identifizieren.

[0065] Die Zugriffskomponente **908** ist konfiguriert, zu bestimmen, ob ein Konto Zugriff auf ein gemeinsam genutztes Betriebsmittel eines anderen Kontos hat. Falls z. B. ein erstes Konto den Zugriff auf ein Betriebsmittel eines anderen, zweiten Kontos anfordert, kann die Zugriffskomponente **908** bestimmen, ob das zweite Konto den Zugriff auf das erste Konto bewilligt hat. Die Zugriffskomponente **908** kann durch das Bestimmen, ob ein Mitbenutzungsobjekt das anfordernde Konto identifiziert, bestimmen, ob ein anforderndes Konto Zugriff hat. Die Zugriffskomponente **908** kann z. B. prüfen, ob das anfordernde Konto in einer Liste von Konten vorhanden ist, die durch ein Mitbenutzungsobjekt gespeichert ist. Die Zugriffskomponente **908** kann außerdem prüfen, ob das Mitbenutzungsobjekt, das das anfordernde Konto identifiziert, Zugriffsrechte (z. B. Bewilligungen) auf das als Ziel gesetzte Datenbetriebsmittel im Mitbenutzerkonto aufweist.

[0066] In einer Ausführungsform kann die Zugriffskomponente **908** das Vorhandensein einer Bewilligung von einer Mitbenutzungsrolle in einem Mitbenutzerkonto für eine anfordernde Rolle im anfordernden Konto prüfen. Die Zugriffskomponente **908** kann prüfen, ob eine Verkettung zwischen einem Aliasobjekt, an das eine Datenbankanforderung oder -anweisung gerichtet war, vorhanden ist, oder ob eine Bewilligung zwischen einer anfordernden Rolle und dem Aliasobjekt vorhanden ist. Die Zugriffskomponente **908** kann z. B. die Existenz oder das Vorhandensein von einem oder mehreren von L1, G1 und G2, die in **Fig. 8** veranschaulicht sind, überprüfen. Weiterhin kann die Zugriffskomponente **908** bezüglich einer Bewilligung zwischen einer Rolle in einem Mitbenutzungsobjekt und einem Objekt (z. B. einer Tabelle oder einer Datenbank) des Mitbenutzerkon-

tos prüfen. Die Zugriffskomponente **908** kann z. B. bezüglich des Vorhandenseins einer Bewilligung zwischen der Rolle **R4** und der Datenbank **D2** in **Fig. 8** prüfen. Falls die Zugriffskomponente **908** bestimmt, dass das anfordernde Konto Zugriff auf das gemeinsam benutzte Betriebsmittel hat, kann die Anforderung durch die Mitbenutzungskomponente **210** oder eine Verarbeitungskomponente **910** erfüllt werden. Falls die Zugriffskomponente **908** bestimmt, dass das anfordernde Konto keine Rechte an den angeforderten Daten oder dem angeforderten Objekt aufweist, wird die Abfrage abgelehnt.

[0067] Die Verarbeitungskomponente **910** ist konfiguriert, Datenbankanforderungen, -abfragen oder -anweisungen zu verarbeiten. Die Verarbeitungskomponente **910** kann eine Anforderung von einem Konto, auf Daten oder Dienste in einem weiteren Konto zuzugreifen oder diese zu nutzen, verarbeiten und eine Antwort auf diese Anforderung bereitstellen. In einer Ausführungsform stellt die Verarbeitungskomponente **910** durch das Verarbeiten der Anforderung unter Verwendung der ursprünglichen Daten in einem Mitbenutzerkonto, das sich vom anfordernden Konto unterscheidet, eine Antwort auf eine Anforderung bereit. Die Anforderung kann z. B. an eine Datenbank oder eine Tabelle gerichtet sein, die in einem oder für ein erstes Konto gespeichert ist, wobei die Verarbeitungskomponente **910** die Anforderung unter Verwendung der Datenbank oder der Tabelle des ersten Kontos verarbeiten und eine Antwort an ein anforderndes, zweites Konto zurückgeben kann.

[0068] In einer Ausführungsform führt die Verarbeitungskomponente **910** die Verarbeitung gemeinsam benutzter Daten aus, ohne eine doppelte Tabelle oder eine andere Datenquelle im anfordernden Konto zu erzeugen. Im Allgemeinen müssen die Daten zuerst in ein Konto aufgenommen werden, das diese Daten verarbeiten oder Operationen mit den Daten ausführen möchte. Die Verarbeitungskomponente **910** kann Verarbeitungszeit, eine Verzögerung und/oder Speicherbetriebsmittel einsparen, indem sie einem Zielkonto ermöglicht, auf die gemeinsam benutzten Betriebsmittel in einem Mitbenutzerkonto zuzugreifen, ohne eine Kopie eines Datenbetriebsmittels im Zielkonto zu erzeugen.

[0069] Die Verarbeitungskomponente **910** kann die Verarbeitung derselben Daten unter Verwendung unterschiedlicher Verarbeitungsbetriebsmittel für verschiedene Konten ausführen. Ein erstes virtuelles Lager für ein erstes Konto kann z. B. eine Datenbankabfrage oder -anweisung unter Verwendung von Daten verarbeiten, die durch ein Mitbenutzerkonto gemeinsam benutzt werden, während ein zweites virtuelles Lager für ein zweites Konto oder das Mitbenutzerkonto eine Datenbankabfrage oder -anweisung unter Verwendung der gemeinsam benutzten Daten des Mitbenutzerkontos verarbeiten kann. Das Verwenden getrennter Verarbeitungsbetriebsmittel, um dieselben Daten zu verarbeiten, kann eine Konkurrenz um die Verarbeitungsbetriebsmittel zwischen den Konten verhindern. Die Verarbeitungsbetriebsmittel können dynamisch bereitgestellte Verarbeitungsbetriebsmittel enthalten. In einer Ausführungsform wird das Verarbeiten der gemeinsam benutzten Daten unter Verwendung eines virtuellen Lagers für das anfordernde Konto ausgeführt, selbst wenn sich die Daten in einem Speicher für ein anderes Konto befinden können.

[0070] Die Komponente **912** für sichere Sichten ist konfiguriert, eine sichere Sicht für ein Mitbenutzungsobjekt, ein Datenfeld eines Mitbenutzungsobjekts, ein Datenfeld eines Datenbankobjekts usw. zu definieren. In einer Ausführungsform definiert die Komponente **912** für sichere Sichten die sichere Sicht unter Verwendung eines SICHER-Schlüsselworts in einem Sichtfeld, wobei sie die SICHER-Eigenschaft in einer Sicht unter Verwendung eines SICHT-ÄNDERN-Befehls setzen oder aufheben kann. In verschiedenen Ausführungsformen kann die Komponente **912** für sichere Sichten derartige Befehle nur auf die manuelle Anordnung eines Klientenkontos implementieren oder konfiguriert sein, derartige Befehle automatisch zu implementieren. Die Komponente **912** für sichere Sichten kann den Parser ändern, um das sichere Schlüsselwort vor dem Sichtnamen und der neuen Regel zum Ändern der Sicht zu unterstützen. In einer Ausführungsform kann die Regel zum Ändern der Sicht allgemeiner sein, um weitere Attribute auf Sichtebene einzubeziehen. Hinsichtlich der Unterstützung von Metadaten können die Sichten effektiv als Tabellen gespeichert sein, wobei die Änderung das Ändern eines Tabellendaten-Beständigkeitsobjekts einschließen kann, das einen Sicher-Merker enthält, der angibt, ob die Sicht eine sichere Sicht ist (dies kann zusätzlich zu dem Sichttext, der das Sicher-Etikett enthält, implementiert sein). Die Definition einer sicheren Sicht (d. h., das Tabellendaten-Beständigkeitsobjekt) kann vor den Anwendern verborgen sein, die nicht der Eigentümer der Sicht sind. In einer derartigen Ausführungsform gibt ein Befehl, um die Sichten zu zeigen, die Ergebnisse wie üblich an den Eigentümer der Sicht zurück, wobei er aber die Definition einer sicheren Sicht nicht an ein zweites Konto eines Nicht-Eigentümers, das Zugriff auf die Sicht hat, zurückgibt.

[0071] Die Komponente **912** für sichere Sichten kann Transformationen eines Parse-Baums ändern, z. B. das Verschmelzen von Sichten und Prädikatinformationen. Die kanonische Implementierung kann das Kommentieren von Abfrageblöcken enthalten, so dass die Abfrageblöcke als von der sicheren Sicht kommend bezeichnet werden. In einer derartigen Implementierung können die Abfrageblöcke nicht mit externen Abfrageblöcken (z. B. das Verschmelzen

von Sichten) oder Ausdrücken (z. B. über eine Filter-Einkellerung) kombiniert werden.

[0072] Die Komponente **912** für sichere Sichten kann den Abfrageplanbaum während der Optimierung, z. B. während eines Filter-Hochziehens und/oder einer Filter-Einkellerung, neu schreiben. Die Komponente **912** für sichere Sichten kann konfiguriert sein, um sicherzustellen, dass kein Ausdruck, der nicht von einer sicheren Sicht stammt, abwärts unter die Sichtgrenzen geschoben werden kann. Die Komponente **912** für sichere Sichten kann konfiguriert sein, um dies durch das Implementieren eines neuen Typs einer Projektion zu erreichen, die sich zu einer Standardprojektion völlig gleich verhält, aber, weil sie keine Standardprojektion ist, irgendeiner der Voraussetzungen für die Neuschreibregel nicht entspricht. Im Ergebnis werden die entsprechenden Neuschreibvorgänge nicht angewendet. Die Komponente **912** für sichere Sichten kann konfiguriert sein, zu identifizieren, welcher Typ von Projektion erzeugt werden soll (z. B. eine Standardprojektion oder eine sichere Projektion), nachdem die Abfrageblöcke als von einer Definition einer sicheren Sicht kommend bestimmt worden sind oder nicht.

[0073] Die Komponente **912** für sichere Sichten ist konfiguriert, die Leistung für sichere Sichten in einem System zum gemeinsamen Benutzen von Daten ohne Kopieren zu optimieren. In verschiedenen in der Technik bekannten Ausführungsformen ist bekannt, dass sichere Sichten einen Leistungsverlust verursachen, der den Optimierer beim Anwenden bestimmter Transformationen wirksam lähmen kann. Derartige Ausführungsformen könnten verbessert werden, indem bestimmte Transformationen für sicher gehalten werden, wobei die Sicherheit angibt, dass die Operationen, die transformiert werden, keine Nebenwirkungen auf das System aufweisen. Derartige Nebenwirkungen können durch eine anwenderdefinierte Funktion (UDF) verursacht werden, die Operationen, die unsichere Operationen nicht leicht identifizieren können, oder Operationen, die fehlschlagen und Informationen über den Datenwert enthüllen können, der die Störung verursacht hat (z. B. wenn durch null geteilt wird oder eine ähnlichen Operation), ausführt. Die Komponente **912** für sichere Sichten kann Ausdrücke mit den Sicherheitseigenschaften des Ausdrucks kommentieren und dann Transformationen ermöglichen, die es ermöglichen, dass ein Ausdruck durch eine Grenze der sicheren Sicht geschoben wird, falls der Ausdruck für sicher gehalten wird. Der Ausdruck kann für sicher gehalten werden, wenn bekannt ist, dass der Ausdruck keine Fehler erzeugt, und der Ausdruck keine anwenderdefinierte Funktion (UDF) enthält. Die Komponente **912** für sichere Sichten kann unter Verwendung eines Rahmens von Ausdruckseigenschaften, in dem die Ausdruckseigenschaften eine Angabe speichern, ob ein Ausdruck Fehler erzeugen kann, bestimmen, ob der Ausdruck Fehler erzeugt.

[0074] **Fig. 10** ist ein schematischer Ablaufplan, der ein beispielhaftes Verfahren **1000** zum Implementieren sicherer Sichten beim gemeinsamen Benutzen von Daten ohne Kopieren in einem mandantenfähigen Datenbanksystem veranschaulicht. Das Verfahren **1000** kann durch ein Datenbanksystem, wie z. B. durch eine Mitbenutzungskomponente **210** wie in den **Fig. 1** oder **Fig. 9**, oder durch ein Datenbanksystem wie in den **Fig. 1**, **Fig. 2** oder **Fig. 3**, ausgeführt werden.

[0075] Das Verfahren **1000** beginnt, wobei eine kontoübergreifende Rechtekomponente **902** bei 1002 ein Mitbenutzungsobjekt in einem ersten Konto, das eine Mitbenutzungsrolle umfasst, erzeugt. Die Komponente **912** für sichere Sichten ordnet bei 1004 die Sichtprivilegien dem Mitbenutzungsobjekt zu, so dass eine zugrundeliegende Einzelheit des Mitbenutzungsobjekts eine Definition einer sicheren Sicht umfasst. Die zugrundeliegende Einzelheit des Mitbenutzungsobjekts kann ein oder mehrere Datenfelder einer zugrundeliegenden Datenbanktabelle, ein Strukturelement einer zugrundeliegenden Datenbanktabelle, eine Datenmenge in einer zugrundeliegenden Datenbanktabelle usw. umfassen. Die kontoübergreifende Rechtekomponente **902** bewilligt bei 1006 einem zweiten Konto kontenübergreifende Zugriffsrechte auf die Mitbenutzungsrolle oder das Mitbenutzungsobjekt im ersten Konto. Eine Anforderungskomponente **906** empfängt bei 1008 eine Anforderung des zweiten Kontos, um auf Daten oder Dienste des ersten Kontos zuzugreifen. Eine Verarbeitungskomponente **910** stellt bei 1010 basierend auf den Daten oder Diensten des ersten Kontos eine Antwort an das zweite Konto bereit. Das Verfahren ist so, dass das Datenfeld des Mitbenutzungsobjekts, das die Definition einer sicheren Sicht umfasst, vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist (siehe 1012).

[0076] **Fig. 11** veranschaulicht einen Prozessablauf **1100** für einen Abfrageplan unter Verwendung einer sicheren anwenderdefinierten Funktion (UDF). In einer Ausführungsform, die UDFs enthält, kann ein Klientenkonto eine sichere UDF definieren. Die Definition der sicheren UDF kann vor den Anwendern, die nicht der Eigentümer der Funktion sind, verborgen sein. In einer Ausführungsform erfordert dies Änderungen an mehreren Operationen. Eine „FUNKTIONEN-ZEIGEN“-Operation gibt z. B. die Ergebnisse wie üblich an den Eigentümer der Funktion zurück, gibt aber die Funktionsdefinition nicht an einen Nicht-Eigentümer, der Zugriff auf die Sicht hat, zurück. Ferner gibt z. B. eine „DDL-ERHALTEN“-Operation die DDL für eine sichere Funktion nicht an einen Nicht-Eigentümer zurück. Ferner gibt z. B. eine „INFORMATIONSSCHEMAFUNKTIONEN“-Operation die Funk-

tionsdefinition nur an die Anwender mit der Eigentümerrolle zurück.

[0077] In einer Ausführungsform erfordern die Javascript-UDFs keine Änderungen an der Verarbeitung. In einer Ausführungsform für SQL-UDFs, die auf Datenbanktabellen zugreifen, können die Verfahren und Systeme sicherstellen, dass keine Optimierungen erlauben, dass vor irgendeiner Filterung, die in der UDF stattfindet, die sich aus Filterprädikaten oder Verknüpfungen ergeben kann, potentiell gefährliche Anwenderausdrücke ausgewertet werden. Eine sichere Sicht kann dies durch das Einführen eines „SecureProjection"-Knotens über dem Plan der Sicht in den Abfrageplanbaum und das Verhindern, dass potentiell gefährliche Anwenderausdrücke durch die SecureProjection geschoben werden, erreichen. In einer Ausführungsform, in der vorhandene Optimierer-Neuschreibvorgänge nicht wissen, was ein SecureProjection-Knoten ist, kann der Knoten alle Optimierungen deaktivieren, die Ausdrücke in den Plan der Sicht verschieben könnten. In einer Ausführungsform kann nur die sichere Optimierung, die zurück hinzugefügt wird, irgendetwas durch diese Barriere verschieben.

[0078] In einer Ausführungsform unterscheiden sich die sicheren UDFs auf zwei Arten von den sicheren Sichten. Erstens kann für sichere UDFs die Barriere auf der Planherangehensweise nicht arbeiten, weil die Dekorrelationsregeln beim Vorhandensein des SecureProjection-Knotens fehlschlagen können. In einer Ausführungsform, in der die Dekorrelation fehlschlägt, kann die Abfrage nicht ausgewertet werden. Zweitens können im Fall einer korrelierten UDF gefährliche Anwenderausdrücke von oben (ähnlich zu den Sichten) oder von „links", d. h., dem Teil der Abfrage, von dem die Korrelation ausgeht, in die UDF geschoben werden.

[0079] In einer Ausführungsform werden korrelierte Unterabfragen und Sichten in einem Planbaum unter Verwendung eines Anwendungsknotens dargestellt. Semantisch wertet der Anwendungsknoten den korrelierten Ausdruck der rechten Eingabe in jeder Zeile der linken Eingabe aus und gibt alle von der rechten Seite zurückgegebenen Zeilen zurück. Die skalar korrelierten Unterabfragen und die korrelierte Sicht werden umgesetzt, um schließlich eine Anwendung in eine Verknüpfung umzusetzen. Wenn die Auswertung der Anwendung zur Laufzeit nicht unterstützt wird, schlägt eine Abfrage, die nicht erfolgreich alle Korrelationen entfernt (und die Anwendung in eine Verknüpfung umsetzt), mit einem Anwenderfehler fehl. Die Dekorrelation kann durch einen Satz von Neuschreibregeln erreicht werden, die etwa dem Baum oberhalb der Anwendung entsprechen, bis der Baum unterhalb der Anwendung nicht länger mit der linken Seite korreliert ist. Dieses Hochziehen von Knoten kann verursachen, dass die Verwendung einer Herangehensweise einer sicheren Sicht des Anordnens einer SecureProjection oben auf dem Baum der Funktion nicht funktioniert. In einer Ausführungsform kann das Anordnen einer SecureProjection über der Anwendung die Dekorrelationsregeln nicht beeinträchtigen, wobei es aber nicht verhindert, dass sich Anwenderausdrücke von der linken Seite der Anwendung in die UDF bewegen.

[0080] In **Fig. 11** veranschaulicht der Prozessablauf **1100** ein Beispiel der Dekorrelation einer tabellenwertigen UDF. Der Prozessablauf **1100** (vor der Dekorrelation) enthält eine Anwendung **1102**, die zu einer Tabellenabtastung **1104** und einem Aggregat **1106** führt. Das Aggregat **1108** führt zu einer Operation eines Filters **1108** und einer Operation einer Verknüpfung **1110**. Die Operation der Verknüpfung **1110** führt zu einer Operation eines Filters **1112** und einer Tabellenabtastung **1116** und außerdem zu einer Operation einer Tabellenabtastung **1114**. In einem Prozessablauf **1100**, wie er in **Fig. 11** veranschaulicht ist, wird die Abfrage durch zwei Dekorrelationsregeln transformiert, um den Knoten der Anwendung **1102** zu entfernen. Die erste Regel verschiebt einen Knoten eines nicht skalaren Aggregats **1106** über die Anwendung **1102**.

[0081] **Fig. 12** veranschaulicht einen Prozessablauf **1200**, bei dem eine zweite Regel einen Anwendungsknoten plus einen korrelierten Filter auf der rechten Seite in eine Verknüpfung umsetzt. Der Prozessablauf **1200** enthält eine Operation eines Aggregats **1202**, die zu der Anwendung **1204** führt. Die Anwendung **1204** fließt zu einer Projektion **1206** und einer Tabellenabtastung **1208** und führt weiter zu einer Operation eines Filters **1210**. Der Filter **1210** führt zu einer Verknüpfung **1212**, die zu einem Filter **1214** und einer Tabellenabtastung **1218** oder alternativ zu einer Tabellenabtastung **1216** fließt.

[0082] **Fig. 13** veranschaulicht einen Prozessablauf **1300** zum Aufdecken von Daten aus einer tabellenwertigen UDF. In einer Ausführungsform muss, um eine UDF sicher zu machen, verhindert werden, dass sich potentiell unsichere Ausdrücke über die Anwendung oder von der linken Seite der Anwendung zur rechten Seite der Anwendung bewegen. In dem in **Fig. 13** veranschaulichten Prozessablauf **1300** führt das Aggregat **1302** zu einer Verknüpfung **1304**, wobei die Verknüpfung **1304** zu einer Projektion **1306** und einer Tabellenabtastung **1308** oder alternativ zu einer Verknüpfung **1310** führt. Die Verknüpfung **1310** führt zu einem Filter **1312** und einer Tabellenabtastung **1316** oder alternativ zu einer Tabellenabtastung **1314**.

[0083] In einer in **Fig. 13** veranschaulichten Ausführungsform ist die Tabellenfunktion vorgesehen, die Daten nur für die Mitarbeiter z. B. aus einer Abteilung einer Mitarbeiterdatenbank aufzudecken. In einem

Beispiel kann die Abfrage verwendet werden, um zu bestimmen, ob es in einer Datenbank irgendwelche Mitarbeiter innerhalb eines spezifischen Gehaltsbereichs gibt. Wenn es einen Mitarbeiter mit einem Gehalt gibt, das größer als ein definierter Schwellenwert ist, erzeugt die Abfrage einen Fehler. Dies kann aufgrund einer Filterprädikat-Einkellerung auftreten. Nachdem die Anwendung in eine Verknüpfung umgesetzt worden ist, kann der Filter unter die beiden Verknüpfungen geschoben werden.

[0084] Fig. 14 veranschaulicht einen Prozessablauf **1400** für einen sicheren Plan für eine tabellenwertige UDF. Der Prozessablauf **1400** für eine sichere tabellenwertige UDF muss vielleicht verhindern, dass sich Ausdrücke von der linken Seite der Anwendung **1404** und oberhalb der Anwendung **1404** in die rechte (UDF-) Seite der Anwendung **1404** bewegen. Um dies zu erreichen, wird die Verwendung von zwei Knoten einer sicheren Projektion verwendet (siehe **1406** und **1402**), wobei eine sichere Projektion **1406** über dem linken Eingang in die Anwendung **1404** angeordnet ist und die andere sichere Projektion **1402** über der Anwendung **1404** angeordnet ist. Zusätzlich können irgendwelche Filterprädikate in der Tabellenfunktion über der Sicheren Projektion **1402** angeordnet sein. Diese Änderungen können während der Übersetzung des Abfrageblocks ausgeführt werden, wenn eine seitliche Sicht, die von einer sicheren Tabellen-UDF ausgeht, in eine Anwendung umgesetzt wird. Der Prozessablauf **1400** enthält die sichere Projektion **1402** über der Anwendung **1404**. Die linke Seite der Anwendung **1404** führt zu der sicheren Projektion **1406** und einer Tabellenabtastung **1408**. Die rechte Seite (UDF-Seite) der Anwendung **1404** enthält das Aggregat **1410**, den Filter **1412** und die Verknüpfung **1414**. Die Verknüpfung **1414** führt zu einem Filter **1416** und einer Tabellenabtastung **1420** oder alternativ zu einer Tabellenabtastung **1418**.

[0085] Fig. 15 veranschaulicht einen Prozessablauf **1500**, nachdem eine nicht skalare Aggregatanwendungsregel angewendet worden ist. Eine sichere Projektion **1502** und ein Aggregat **1504** sind über der Anwendung **1506** angeordnet. Die linke Seite der Anwendung **1506** enthält eine Projektion **1508**, eine sichere Projektion **1510** und eine Tabellenabtastung **1512**. Die rechte Seite (UDF-Seite) der Anwendung **1506** enthält einen Filter **1514** und eine Verknüpfung **1516**. Die Verknüpfung **1516** enthält einen Filter **1518** und eine Tabellenabtastung **1522** und führt alternativ zu einer Tabellenabtastung **1520**.

[0086] Fig. 16 veranschaulicht einen Prozessablauf **1600**, nachdem eine Umsetzungsanwendung auf eine Verknüpfungsregel angewendet worden ist. Die sichere Projektion **1602** und das Aggregat **1604** sind über der Verknüpfung **1606** angeordnet. Die linke Seite der Verknüpfung **1606** enthält eine Projektion **1608**, eine sichere Projektion **1610** und eine Tabellenabtastung **1612**. Die rechte Seite der Verknüpfung **1606** enthält eine Verknüpfung **1614**. Die Verknüpfung **1614** führt zu einem Filter **1616** und einer Tabellenabtastung **1620** oder alternativ zu einer Tabellenabtastung **1618**.

[0087] In einer Ausführungsform verweigert eine sichere Projektion, dass potentiell unsichere Ausdrücke durch die Projektion nach unten geschoben werden, und verhindert, dass alle Ausdrücke über die Projektion hochgezogen werden. Die unsicheren Ausdrücke auf der linken Seite können nie aus der linken Seite des Plans kommen, wobei die unsicheren Ausdrücke oberhalb der Anwendung nicht nach unten unter die Verknüpfung geschoben werden, die schließlich die Anwendung ersetzt.

[0088] Fig. 17 veranschaulicht einen Prozessablauf **1700** vor der Dekorrelation. Der Prozessablauf **1700** enthält einen Filter **1702** und eine sichere Projektion **1704**, die über der Anwendung **1706** angeordnet ist. Die linke Seite der Anwendung **1706** enthält eine sichere Projektion **1708** und eine Tabellenabtastung **1710**. Die rechte Seite (UDF-Seite) der Anwendung **1706** enthält ein Aggregat **1712**, einen Filter **1714** und eine Verknüpfung **1716**. Die Verknüpfung **1716** enthält einen Filter **1718** und eine Tabellenabtastung **1722** und enthält alternativ eine Tabellenabtastung **1720**. Weil sich der Filter oberhalb einer sicheren Projektion befindet, kann er nicht nach unten unter die Grenze der UDF geschoben werden.

[0089] Fig. 18 veranschaulicht einen Prozessablauf **1800** vor der Dekorrelation. Der Prozessablauf **1800** enthält eine sichere Projektion **1802** über der Anwendung **1804**. Die linke Seite der Anwendung **1804** enthält eine sichere Projektion **1806**, einen Filter **1808** und eine Tabellenabtastung **1810**. Die rechte Seite (UDF-Seite) der Anwendung **1804** enthält ein Aggregat **1812**, einen Filter **1814** und eine Verknüpfung **1816**. Die Verknüpfung **1816** enthält einen Filter **1818** und eine Tabellenabtastung **1822** und führt alternativ zu einer Tabellenabtastung **1820**. Weil sich der Filter unterhalb einer sicheren Projektion befindet, kann er nicht nach oben gezogen und auf die rechte Seite der Anwendung **1804** geschoben werden.

[0090] In einer Ausführungsform kann eine skalare UDF während der UDF-Erweiterung durch einige Unterabfragen ersetzt sein, die später während des Neuschreibens der Abfrage vor der Dekorrelation in eine Anwendung umgesetzt werden. Ferner kann ein früher Unterabfrage-Neuschreibdurchgang ausgeführt werden, bevor die Abfrage in ihre Plandarstellung umgesetzt wird. Die Entschachtelung der sicheren Unterabfragen kann an diesem Punkt verweigert werden, so dass es keine Notwendigkeit gibt, die sichere Dekorrelation an einer Stelle handzuhaben. Die Knoten einer sicheren Projektion über und

links von der Anwendung für Unterabfragen, die von sicheren skalaren UDFs kommen.

[0091] In einer Ausführungsform verhindert das Anordnen einer sicheren Projektion über der linken Eingabe in die Anwendung ein Neuordnen der Verknüpfung zwischen der linken Eingabe und der von der Anwendung erzeugten Verknüpfung. Das Anordnen einer sicheren Projektion über der Anwendung kann gleichermaßen das Neuordnen der Verknüpfung oberhalb und unterhalb der Anwendung verhindern. Eine Alternative kann das Implementieren spezifischer Dekorrelationsregeln enthalten, die für sichere UDFs gelten.

[0092] Fig. 19 veranschaulicht einen Prozessablauf **1900** für eine Abfrage, wie sie im Abfrageprofil erscheint. Das Abfrageprofil deckt keine Einzelheiten dessen auf, was eine sichere SQL-UDF ausführt, einschließlich der Tabellen, auf die zugegriffen wird, der Menge der verarbeiteten Daten usw. Eine sichere Sicht wird im Abfrageprofil als eine sichere Einzelknoten-Sicht angezeigt, wobei keine der Einzelheiten des zugrundeliegenden Plans aufgedeckt werden. Der Prozessablauf **1900** enthält eine sichere anwenderdefinierte Funktion **1902**, die zu einem Filter **1904** und einer Tabellenabtastung **1906** führt. Die Seiten der Abfragehistorie und der Abfrageeinzelheiten der Anwenderschnittstelle können Informationen über die durch die Abfrage abgetasteten Bytes und Partitionen enthalten. Wenn eine Abfrage eine sichere Sicht enthält, werden diese Informationen nicht an die Anwenderschnittstelle gesendet, wenn sie die Abfragewarnungsstatistik anfordert. Die Abfragen, die sichere UDFs enthalten, können in der gleichen Weise behandelt werden.

[0093] Fig. 20 ist ein Blockschaltplan, der eine beispielhafte Rechenvorrichtung **2000** zeigt. In einigen Ausführungsformen wird die Rechenvorrichtung **2000** verwendet, um eines oder mehrere der hier erörterten Systeme und eine oder mehrere der hier erörterten Komponenten zu implementieren. Ferner kann die Rechenvorrichtung **2000** mit irgendeinem der hier beschriebenen Systeme und irgendeiner der hier beschriebenen Komponenten wechselwirken. Entsprechend kann die Rechenvorrichtung **2000** verwendet werden, um verschiedene Prozeduren und Aufgaben, wie z. B. jene, die hier erörtert sind, auszuführen. Die Rechenvorrichtung **2000** kann als ein Server, ein Client oder irgendeine andere Rechenentität arbeiten. Die Rechenvorrichtung **2000** kann irgendeine einer umfassenden Vielfalt von Rechenvorrichtungen sein, wie z. B. ein Desktop-Computer, ein Notebook-Computer, ein Server-Computer, ein Handheld-Computer, ein Tablet-PC und dergleichen.

[0094] Die Rechenvorrichtung **2000** enthält einen oder mehrere Prozessor(en) **2002**, eine oder mehrere Speichervorrichtung(e) **2004**, eine oder mehrere Schnittstelle(n) 2006, eine oder mehrere Massenspeichervorrichtung(en) **2008** und eine oder mehrere Eingabe-/Ausgabe- (E/A-) Vorrichtung(en) **2010**, von denen alle an einen Bus **2012** gekoppelt sind. Der (die) Prozessor(en) **2002** enthält (enthalten) einen oder mehrere Prozessoren oder Controller, die die in der (den) Speichervorrichtung(en) **2004** und/oder der (den) Massenspeichervorrichtung(en) **2008** gespeicherten Anweisungen ausführen. Der (die) Prozessor(en) **2002** kann (können) außerdem verschiedene Typen von computerlesbaren Medien, wie z. B. einen Cache-Speicher, enthalten.

[0095] Die Speichervorrichtung(en) **2004** enthält (enthalten) verschiedene computerlesbare Medien, wie z. B. flüchtigen Speicher (z. B. Schreib-Lese-Speicher (RAM)) und/oder nichtflüchtigen Speicher (z. B. Festwertspeicher (ROM)). Die Speichervorrichtung(en) **2004** kann (können) außerdem wiederbeschreibbaren ROM, wie z. B. Flash-Speicher, enthalten.

[0096] Die Massenspeichervorrichtung(en) **2008** enthält (enthalten) verschiedene computerlesbare Medien, wie z. B. Magnetbänder, Magnetplatten, optische Platten, Festkörperspeicher (z. B. Flash-Speicher) usw. In der (den) Massenspeichervorrichtung (en) **2008** können außerdem verschiedene Laufwerke enthalten sein, um das Lesen von den und/oder das Schreiben in die verschiedenen computerlesbaren Medien zu ermöglichen. Die Massenspeichervorrichtung(en) **2008** enthält (enthalten) abnehmbare Medien und/oder nicht abnehmbare Medien.

[0097] Die E/A-Vorrichtung(en) 2010 enthält (enthalten) verschiedene Vorrichtungen, die es ermöglichen, dass Daten und/oder andere Informationen in die Computervorrichtung **2000** eingegeben oder aus der Computervorrichtung **2000** wiedergewonnen werden. Eine beispielhafte E/A-Vorrichtung(en) 2010 enthält (enthalten) Cursor-Steuervorrichtungen, Tastaturen, Tastenfelder, Mikrophone, Monitore oder andere Anzeigevorrichtungen, Lautsprecher, Drucker, Netzschnittstellenkarten, Modems, Objektive, CCDs oder andere Bilderfassungsvorrichtungen und dergleichen.

[0098] Die Schnittstelle(n) 2006 enthält (enthalten) verschiedene Schnittstellen, die es der Rechenvorrichtung **2000** ermöglichen, mit anderen Systemen, Vorrichtungen oder Rechenumgebungen zu wechselwirken. Eine beispielhafte Schnittstelle(n) 2006 enthält (enthalten) irgendeine Anzahl verschiedener Netzschnittstellen, wie z. B. Schnittstellen zu lokalen Netzen (LANs), Weitbereichsnetzen (WANs), drahtlosen Netzen und dem Internet.

[0099] Der Bus **2012** ermöglicht es sowohl dem (den) Prozessor(en) **2002**, der (den) Speichervor-

richtung(en) **2004**, der (den) Schnittstelle(n) 2006, der (den) Massenspeichervorrichtung(en) **2008** und der (den) E/A-Vorrichtung(en) 2010, miteinander als auch mit anderen Vorrichtungen oder Komponenten, die an den Bus **2012** gekoppelt sind, zu kommunizieren. Der Bus **2012** repräsentiert einen oder mehrere von mehreren Typen von Busstrukturen, wie z. B. einen Systembus, einen PCI-Bus, einen IEEE 1394-Bus, einen USB-Bus usw.

[0100] Für Veranschaulichungszwecke sind Programme und andere ausführbare Programmkomponenten hier als diskrete Blöcke gezeigt, obwohl erkannt wird, dass sich derartige Programme und Komponenten zu verschiedenen Zeiten in verschiedenen Speicherkomponenten der Rechenvorrichtung **2000** befinden können und durch den (die) Prozessor(en) **2002** ausgeführt werden. Alternativ können die hier beschriebenen Systeme und Prozeduren in Hardware oder einer Kombination aus Hardware, Software und/oder Firmware implementiert sein. Es können z. B. eine oder mehrere anwendungsspezifische integrierte Schaltungen (ASICs) programmiert sein, um eines oder mehrere der Systeme und eine oder mehrere der Prozeduren, die hier beschrieben sind, auszuführen. Wie die Begriffe „Modul“ oder „Komponente“ hier verwendet werden, ist vorgesehen, dass sie die Implementierungsvorrichtung zum Ausführen eines Prozesses, wie z. B. durch Hardware oder eine Kombination aus Hardware, Software und/oder Firmware, für die Zwecke des Ausführens aller oder Teile der hierin offenbarten Operationen übermitteln.

Beispiele

[0101] Die folgenden Beispiele betreffen weitere Ausführungsformen.

[0102] Das Beispiel 1 ist ein Verfahren zum Bereitstellen sicherer Sichten für gemeinsam benutzte Daten in einer mandantenfähigen Datenbank. Das Verfahren enthält das Erzeugen eines Mitbenutzungsobjekts in einem ersten Konto, das eine Mitbenutzungsrolle umfasst. Das Verfahren enthält das Zuordnen von Sichtprivilegien dem Mitbenutzungsobjekt, so dass eine zugrundeliegende Einzelheit des Mitbenutzungsobjekts eine Definition einer sicheren Sicht umfasst. Das Verfahren enthält das Bewilligen kontenübergreifender Zugriffsrechte auf die Mitbenutzungsrolle oder das Mitbenutzungsobjekt im ersten Konto einem zweiten Konto. Das Verfahren enthält das Empfangen einer Anforderung vom zweiten Konto, um auf Daten oder Dienste des ersten Kontos zuzugreifen, und das Bereitstellen einer Antwort an das zweite Konto basierend auf den Daten oder Diensten des ersten Kontos. Das Verfahren ist so, dass die zugrundeliegende Einzelheit des Mitbenutzungsobjekts, das die Definition einer sicheren Sicht umfasst, vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist.

[0103] Das Beispiel 2 ist ein Verfahren wie im Beispiel 1, wobei die zugrundeliegende Einzelheit des Mitbenutzungsobjekts eines oder mehrere des Folgenden umfasst: ein Datenfeld einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; ein Strukturelement einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; oder eine Datenmenge in einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts.

[0104] Das Beispiel 3 ist ein Verfahren wie in einem der Beispiele 1-2, wobei die Definition einer sicheren Sicht vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist.

[0105] Das Beispiel 4 ist ein Verfahren wie in einem der Beispiele 1-3, wobei das Mitbenutzungsobjekt umfasst: einen für das erste Konto eindeutigen Objektnamen; eine Objektrolle; und eine Referenzliste, die eine Liste von einem oder mehreren zweiten Konten, die berechtigt sind, auf das Mitbenutzungsobjekt zuzugreifen, umfasst.

[0106] Das Beispiel 5 ist ein Verfahren wie in einem der Beispiele 1-4, wobei das Zuordnen von Sichtprivilegien dem Mitbenutzungsobjekt eines oder mehrere des Folgenden umfasst: Ändern eines Tabellendaten-Beständigkeitsobjekts einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; oder Anbringen eines Sicher-Merkers an einem zugrundeliegenden Datenfeld des Mitbenutzungsobjekts, wobei der Sicher-Merker angibt, dass das zugrundeliegende Datenfeld des Mitbenutzungsobjekts durch eine sichere Sicht geschützt ist.

[0107] Das Beispiel 6 ist ein Verfahren wie in einem der Beispiele 1-5, das ferner das Kommentieren der Anforderung von dem zweiten Konto umfasst, um die Anforderung als von einer sicheren Sicht kommend zu bezeichnen.

[0108] Das Beispiel 7 ist ein Verfahren wie in einem der Beispiele 1-6, das ferner das Einschränken umfasst, dass die Anforderung von dem zweiten Konto mit einem externen Abfrageblock oder -ausdruck verschmolzen wird.

[0109] Das Beispiel 8 ist ein Verfahren wie in einem der Beispiele 1-7, das ferner das Kommentieren der Anforderung von dem zweiten Konto mit einer Sicherheitseigenschaft umfasst, die angibt, ob die Anforderung sicher ist.

[0110] Das Beispiel 9 ist ein Verfahren wie in einem der Beispiele 1-8, wobei die die Anforderung sicher ist, falls: bekannt ist, dass die Anforderung keine Fehler erzeugt; und die Anforderung keine anwenderdefinierte Funktion umfasst.

[0111] Das Beispiel 10 ist ein Verfahren wie in einem der Beispiele 1-9, wobei das Empfangen der Anforde-

rung vom zweiten Konto das Empfangen eines Aliasobjekts, das einen eindeutigen Namen umfasst, umfasst, wobei das Aliasobjekt mit einem obersten Objekt in einer gemeinsam benutzten Objekthierarchie verkettet ist.

[0112] Das Beispiel 11 ist ein Verfahren wie in einem der Beispiele 1-10, wobei das Bereitstellen der Antwort an das zweite Konto das Bewilligen von Aliasobjekt-Benutzungsprivilegien in einer aktivierten Rolle innerhalb des Mitbenutzungsobjekts umfasst.

[0113] Das Beispiel 12 ist ein System zum Bereitstellen sicherer Sichten für gemeinsam benutzte Daten in einer mandantenfähigen Datenbank. Das System enthält Mittel zum Bestimmen, dass ein erstes Konto einem zweiten Konto Zugriff auf ein Datenbankobjekt des ersten Kontos bewilligt hat. Das System enthält Mittel zum Zuordnen von Sichtprivilegien für das Datenbankobjekt des ersten Kontos, so dass eine zugrundeliegende Einzelheit des Datenbankobjekts eine Definition einer sicheren Sicht umfasst. Das System enthält Mittel zum Empfangen einer Anforderung vom zweiten Konto, um auf das Datenbankobjekt des ersten Kontos zuzugreifen. Das System enthält Mittel zum Verarbeiten der Anforderung unter Verwendung eines virtuellen Lagers, das dem zweiten Konto entspricht, wobei das virtuelle Lager einen Satz von einem oder mehreren Computerknoten enthält, die konfiguriert sind, auf Daten in einer Speicherschicht zuzugreifen, die dem ersten Konto entspricht, um eine Antwort auf die Anforderung zu erzeugen. Das System ist so, dass die zugrundeliegende Einzelheit des Datenbankobjekts, das die Definition einer sicheren Sicht umfasst, vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist.

[0114] Das Beispiel 13 ist ein System wie in Beispiel 12, wobei die Mittel zum Bestimmen, dass das erste Konto einem zweiten Konto Zugriff bewilligt hat, Mittel zum Bestimmen umfassen, dass das erste Konto das zweite Konto als Rechte für ein Mitbenutzungsobjekt oder eine Mitbenutzungsrolle in dem ersten Konto aufweisend auflistet, wobei das Mitbenutzungsobjekt oder die Mitbenutzungsrolle Zugriffsrechte auf das Datenbankobjekt des ersten Kontos aufweist.

[0115] Das Beispiel 14 ist ein System wie in einem der Beispiele 12-13, wobei die Mittel zum Empfangen der Anforderung Mittel zum Empfangen einer Anforderung umfassen, die an ein Aliasobjekt im zweiten Konto gerichtet ist, wobei das Aliasobjekt mit dem Datenbankobjekt verkettet ist.

[0116] Das Beispiel 15 ist ein System wie in einem der Beispiele 12-14, wobei die Mittel zum Implementieren von Privilegien für das Datenbankobjekt des ersten Kontos eines oder mehrere des Folgenden umfassen: Mittel zum Ändern eines Tabellendaten-Beständigkeitsobjekts für eine zugrundeliegende Tabelle des Datenbankobjekts; oder Mittel zum Anbringen eines Sicher-Merkers an einem zugrundeliegenden Datenfeld des Datenbankobjekts, wobei der Sicher-Merker angibt, dass das zugrundeliegende Datenfeld des Datenbankobjekts durch eine sichere Sicht geschützt ist.

[0117] Das Beispiel 16 ist ein System wie in einem der Beispiele 12-15, das ferner Mittel zum Kommentieren der Anforderung vom zweiten Konto umfasst, um die Anforderung als von einer sicheren Sicht kommend zu bezeichnen.

[0118] Das Beispiel 17 ist ein System wie in einem der Beispiele 12-16, wobei die zugrundeliegende Einzelheit des Datenbankobjekts eines oder mehrere des Folgenden umfasst: ein Datenfeld einer zugrundeliegenden Tabelle des Datenbankobjekts; ein Strukturelement einer zugrundeliegenden Tabelle des Datenbankobjekts; oder eine Datenmenge in dem Datenbankobjekt.

[0119] Das Beispiel 18 ist ein nicht transitorisches computerlesbares Speichermedium, das Anweisungen speichert, die, wenn sie durch einen oder mehrere Prozessoren ausgeführt werden, den einen oder die mehreren Prozessoren veranlassen: ein Mitbenutzungsobjekt in einem ersten Konto zu erzeugen, das eine Mitbenutzungsrolle umfasst; Sichtprivilegien für das Mitbenutzungsobjekt zuzuordnen, so dass eine zugrundeliegende Einzelheit des Mitbenutzungsobjekts eine Definition einer sicheren Sicht umfasst; einem zweiten Konto kontoübergreifende Zugriffsrechte auf die Mitbenutzungsrolle oder das Mitbenutzungsobjekt in dem ersten Konto zu bewilligen; eine Anforderung von dem zweiten Konto zu empfangen, auf Daten oder Dienste des ersten Kontos zuzugreifen; und eine Antwort an das zweite Konto basierend auf den Daten oder Diensten des ersten Kontos bereitzustellen; wobei die zugrundeliegende Einzelheit des Mitbenutzungsobjekts, das die Definition einer sicheren Sicht umfasst, vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist.

[0120] Das Beispiel 19 ist ein nicht transitorisches computerlesbares Speichermedium wie im Beispiel 18, wobei die zugrundeliegende Einzelheit des Mitbenutzungsobjekts eines oder mehrere des Folgenden umfasst: ein Datenfeld einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; ein Strukturelement einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; oder eine Datenmenge in einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts.

[0121] Das Beispiel 20 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-19, wobei die Definition einer sicheren Sicht vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist.

[0122] Das Beispiel 21 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-20, wobei das Mitbenutzungsobjekt des ersten Kontos eines oder mehrere eines Datenbankobjektes, eines Schemas, einer Tabelle, einer Folge oder einer Funktion umfasst.

[0123] Das Beispiel 22 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-21, wobei die Anweisungen ferner den einen oder die mehreren Prozessoren veranlassen, eine Liste von Objekten zu speichern, für die die Mitbenutzungsrolle Zugriffsrechte aufweist.

[0124] Das Beispiel 23 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-22, wobei die Anweisungen den einen oder die mehreren Prozessoren veranlassen, durch das Speichern einer Angabe des zweiten Kontos in einer Referenzliste von Konten, die Rechte für die Mitbenutzungsrolle oder das Mitbenutzungsobjekt im ersten Konto aufweisen, die kontoübergreifenden Zugriffsrechte zu bewilligen.

[0125] Das Beispiel 24 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-23, wobei die Anweisungen den einen oder die mehreren Prozessoren veranlassen, durch das Empfangen einer an ein Aliasobjekt im zweiten Konto gerichteten Anforderung die Anforderung zu empfangen.

[0126] Das Beispiel 25 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-24, wobei die Anweisungen den einen oder die mehreren Prozessoren veranlassen, durch das Verarbeiten der Anforderung, ohne eine Duplikattabelle oder eine andere Datenquelle im zweiten Konto zu erzeugen, die Antwort bereitzustellen.

[0127] Das Beispiel 26 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-25, wobei das Mitbenutzungsobjekt umfasst: einen Objektnamen, der für das erste Konto eindeutig ist; eine Objektrolle; und eine Referenzliste, die eine Liste von einem oder mehreren zweiten Konten umfasst, die berechtigt sind, auf das Mitbenutzungsobjekt zuzugreifen.

[0128] Das Beispiel 27 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-26, wobei die Anweisungen den einen oder die mehreren Prozessoren veranlassen, Sichtprivilegien für das Mitbenutzungsobjekt durch eines oder mehrere des Folgenden zuzuordnen: Ändern eines Tabellendaten-Beständigkeitsobjekts einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; oder Anbringen eines Sicher-Merkers an einem zugrundeliegenden Datenfeld des Mitbenutzungsobjekts, wobei der Sicher-Merker angibt, dass das zugrundeliegende Datenfeld durch eine sichere Sicht geschützt sein sollte.

[0129] Das Beispiel 28 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-27, wobei die Anweisungen ferner den einen oder die mehreren Prozessoren veranlassen, die Anforderung von dem zweiten Konto zu kommentieren, um die Anforderung als von einer sicheren Sicht kommend zu bezeichnen.

[0130] Das Beispiel 29 ist ein nicht transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-28, wobei die Anweisungen ferner den einen oder die mehreren Prozessoren veranlassen, es einzuschränken, dass die Anforderung von dem zweiten Konto mit einem externen Abfrageblock oder -ausdruck verschmolzen wird.

[0131] Das Beispiel 30 ist ein nicht-transitorisches computerlesbares Speichermedium wie in einem der Beispiele 18-29, wobei die Anweisungen ferner den einen oder die mehreren Prozessoren veranlassen, die Anforderung vom zweiten Konto mit einer Sicherheitseigenschaft zu kommentieren, die angibt, ob die Anforderung sicher ist, und wobei die Anforderung sicher ist, falls: bekannt ist, dass die Anforderung keine Fehler erzeugt; und die Anforderung keine anwenderdefinierte Funktion umfasst.

[0132] Das Beispiel 31 ist eine Vorrichtung, die Mittel zum Ausführen eines Verfahrens oder zum Verwirklichen einer Vorrichtung oder eines Systems wie in einem der Beispiele 1-30 enthält.

[0133] Das Beispiel 32 ist ein maschinenlesbarer Speicher, der maschinenlesbare Anweisungen enthält, um, wenn sie ausgeführt werden, nach einem der Beispiele 1-30 ein Verfahren zu implementieren oder eine Vorrichtung zu verwirklichen.

[0134] Verschiedene Techniken oder bestimmte Aspekte oder Teile davon können die Form von Programmcode (d. h., Anweisungen) annehmen, die in greifbaren Medien, wie z. B. Disketten, CD-ROMs, Festplatten, einem nicht transitorischen computerlesbaren Speichermedium oder irgendeinem anderen maschinenlesbaren Speichermedium, verkörpert sind, wobei, wenn der Programmcode in eine Maschine, wie z. B. einen Computer, geladen und durch diese ausgeführt wird, die Maschine eine Vorrichtung zum Praktizieren der verschiedenen Techniken wird. Im Fall der Ausführung des Programmcodes in programmierbaren Computern kann die Rechenvorrichtung einen Prozessor, ein durch den Prozessor lesbares Speichermedium (einschließlich flüchtiger und nichtflüchtiger Speicher und/oder Speicherelemente), wenigstens eine Eingabevorrichtung und wenigstens eine Ausgabevorrichtung enthalten. Die flüchtigen und nichtflüchtigen Speicher- und/oder Speicher-

elemente können ein RAM, ein EPROM, ein Flash-Laufwerk, ein optisches Laufwerk, eine magnetische Festplatte oder ein anderes Medium zum Speichern elektronischer Daten sein. Ein oder mehrere Programme, die die verschiedenen hier beschriebenen Techniken implementieren oder benutzen können, können eine Anwendungsprogrammierschnittstelle (API), wiederverwendbare Steuerelemente und dergleichen verwenden. Derartige Programme können in einer höheren prozeduralen oder einer objektorientierten Programmiersprache implementiert sein, um mit einem Computersystem zu kommunizieren. Das (die) Programm(e) kann (können) jedoch auf Wunsch in einer Assembler- oder Maschinensprache implementiert sein. In jedem Fall kann die Sprache eine kompilierte oder interpretierte Sprache sein und mit einer Hardware-Implementierung kombiniert sein.

[0135] Es sollte erkannt werden, dass viele der in dieser Beschreibung beschriebenen Funktionseinheiten als eine oder mehrere Komponenten implementiert sein können, was ein Begriff ist, der verwendet wird, um die Unabhängigkeit ihrer Implementierung insbesondere hervorzuheben. Eine Komponente kann z. B. als eine Hardware-Schaltung implementiert sein, die kundenspezifische Schaltungen mit sehr hohem Integrationsgrad (VLSI) oder Gatteranordnungen, handelsübliche Halbleiter, wie z. B. Logikchips, Transistoren oder andere diskrete Komponenten, umfasst. Eine Komponente kann außerdem in programmierbaren Hardware-Vorrichtungen, wie z. B. in feldprogrammierbaren Gatteranordnungen, einer programmierbaren Anordnungslogik, programmierbaren Logikvorrichtungen oder dergleichen, implementiert sein.

[0136] Die Komponenten können außerdem in Software zur Ausführung durch verschiedene Typen von Prozessoren implementiert sein. Eine identifizierte Komponente eines ausführbaren Codes kann z. B. einen oder mehrere physische oder logische Blöcke von Computeranweisungen umfassen, die z. B. als ein Objekt, eine Prozedur oder eine Funktion organisiert sein können. Dennoch müssen die Ausführbaren einer identifizierten Komponente nicht physisch zusammen angeordnet sein, sondern sie können verschiedene, an verschiedenen Orten gespeicherte Anweisungen umfassen, die, wenn sie logisch miteinander verknüpft werden, die Komponente umfassen und den angegebenen Zweck für die Komponente erreichen.

[0137] In der Tat kann eine Komponente des ausführbaren Codes eine einzelne Anweisung oder viele Anweisungen sein, wobei sie sogar über mehrere verschiedene Codesegmente, zwischen verschiedenen Programmen und über mehrere Speichervorrichtungen verteilt sein kann. Ähnlicher können die Betriebsdaten hier innerhalb von Komponenten identifiziert und veranschaulicht sein, wobei sie in jeder geeigneten Form verkörpert und innerhalb irgendeines geeigneten Typs einer Datenstruktur organisiert sein können. Die Betriebsdaten können als ein einziger Datensatz gesammelt sein oder über verschiedene Orte einschließlich über verschiedene Speichervorrichtungen verteilt sein und können wenigstens teilweise lediglich als elektronische Signale in einem System oder Netz vorhanden sein. Die Komponenten können passiv oder aktiv sein, einschließlich Agenten, die betreibbar sind, um die gewünschten Funktionen auszuführen.

[0138] Überall in dieser Beschreibung bedeutet ein Verweis auf „ein Beispiel“, dass ein bestimmtes Merkmal, eine bestimmte Struktur oder eine bestimmte Eigenschaft, das bzw. die im Zusammenhang mit dem Beispiel beschrieben wird, in wenigstens einer Ausführungsform der vorliegenden Offenbarung enthalten ist. Folglich beziehen sich nicht notwendigerweise alle Auftritte des Ausdrucks „in einem Beispiel“ an verschiedenen Stellen überall in dieser Beschreibung auf dieselbe Ausführungsform.

[0139] Mehrere Elemente, Strukturelemente, Kompositionselemente und/oder Materialien, wie sie hier verwendet werden, können der Einfachheit halber in einer gemeinsamen Liste dargestellt sein. Diese Listen sollten jedoch so ausgelegt werden, als ob jedes Element der Liste einzeln als ein separates und eindeutiges Element identifiziert ist. Folglich sollte kein einzelnes Element einer derartigen Liste allein basierend auf seiner Darstellung in einer gemeinsamen Gruppe ohne gegenteilige Angaben als ein de facto Äquivalent zu irgendeinem anderen Element derselben Liste ausgelegt werden. Zusätzlich kann hier auf verschiedene Ausführungsformen und Beispiele der vorliegenden Offenbarung zusammen mit Alternativen für ihre verschiedenen Komponenten Bezug genommen werden. Es wird erkannt, dass derartige Ausführungsformen, Beispiele und Alternativen nicht als de facto Äquivalente zueinander auszulegen sind, sondern als getrennte und autonome Darstellungen der vorliegenden Offenbarung zu betrachten sind.

[0140] Obwohl das Vorstehende für die Zwecke der Klarheit ausführlich beschrieben worden ist, ist es offensichtlich, dass bestimmte Änderungen und Modifikationen vorgenommen werden können, ohne von dessen Prinzipien abzuweichen. Es sollte angegeben werden, dass es viele alternative Arten zum Implementieren sowohl der hier beschriebenen Prozesse als auch der hier beschriebenen Vorrichtungen gibt. Entsprechend sind die vorliegenden Ausführungsformen als veranschaulichend und nicht als einschränkend zu betrachten.

[0141] Die Fachleute auf dem Gebiet erkennen, dass viele Änderungen an den Einzelheiten der oben beschriebenen Ausführungsformen vorgenommen werden können, ohne von den zugrundeliegen-

den Prinzipien der Offenbarung abzuweichen. Der Schutzumfang der vorliegenden Offenbarung sollte daher nur durch die folgenden Ansprüche bestimmt werden.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Patentliteratur

- US 16/694906 [0001]
- US 16/055824 [0001]

Schutzansprüche

1. System, das umfasst: einen Speicher; und einen oder mehrere Prozessoren, die betriebstechnisch an den Speicher gekoppelt sind, wobei der eine oder die mehreren Prozessoren:
Sichtprivilegien einer sicheren Sicht einer oder mehreren von mehreren zugrundeliegenden Einzelheiten eines Mitbenutzungsobjekts eines ersten Kontos zuzuordnen, so dass jede der einen oder der mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts eine Definition der sicheren Sicht umfasst;
einem zweiten Konto kontoübergreifende Zugriffsrechte auf das Mitbenutzungsobjekt des ersten Kontos zu gewähren;
in Reaktion auf das Empfangen einer Anforderung vom zweiten Konto, auf das Mitbenutzungsobjekt zuzugreifen, das Modifizieren eines Abfrageplans der Anforderung unter Verwendung eines sicheren Projektionsoperators, um sicherzustellen, dass alle sicherheitsbezogenen Filterungen, die die Anforderung betreffen, stattfinden, bevor irgendein Ausdruck, der die Anforderung betrifft und sich außerhalb der sicheren Sicht befindet, ausgewertet wird, wobei die Ausdrücke, die nicht von der sicheren Sicht stammen, nicht unter eine Grenze der sicheren Sicht nach unten geschoben werden können, so dass die eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts vor dem zweiten Konto verborgen und für das erste Konto sichtbar sind; und
durch das zweite Konto auf das Mitbenutzungsobjekt zuzugreifen, ohne eine Kopie von irgendeiner der einen oder der mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts im zweiten Konto zu erzeugen.

2. System nach Anspruch 1, wobei der eine oder die mehreren Prozessoren ferner: innerhalb des zweiten Kontos ein Aliasobjekt zu erzeugen, das auf das Mitbenutzungsobjekt verweist; und das Aliasobjekt mit der einen oder den mehreren von den mehreren der zugrundeliegenden Einzelheiten zu verketten, wobei das zweite Konto die Anforderung, auf das eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten zuzugreifen, zu dem Aliasobjekt leitet und unter Verwendung der Zugriffsrechte des Mitbenutzungsobjekts auf das eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten zugreift.

3. System nach Anspruch 2, wobei das Aliasobjekt als ein Proxy für das Mitbenutzungsobjekt dient.

4. System nach Anspruch 1, wobei der eine oder die mehreren Prozessoren, um dem zweiten Konto kontenübergreifende Zugriffsrechte auf das Mitbenutzungsobjekt zu gewähren, das zweite Konto in eine Liste von Konten, die Zugriffsrechte auf das Mitbenutzungsobjekt des ersten Kontos aufweisen, aufnehmen sollen.

5. System nach Anspruch 1, wobei der eine oder die mehreren Prozessoren ferner: jeden Ausdruck der Anforderung von dem zweiten Konto mit den Sicherheitseigenschaften jedes Ausdrucks zu kommentieren; für jeden Ausdruck, der als sicher bestimmt wird, Transformationen zu ermöglichen, die es ermöglichen, dass derjede Ausdruck durch eine Grenze einer sicheren Sicht geschoben wird, wobei ein Ausdruck als sicher bestimmt wird, falls bestimmt wird, dass er keine Fehler erzeugt und keine anwenderdefinierte Funktion (UDF) enthält.

6. System nach Anspruch 1, wobei eine zugrundeliegende Einzelheit des Mitbenutzungsobjekts eines oder mehrere des Folgenden umfasst: ein Datenfeld einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; ein Strukturelement einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; oder eine Datenmenge in dem Mitbenutzungsobjekt.

7. Computerprogramm, das Anweisungen umfasst, die, wenn sie durch einen oder mehrere Prozessoren ausgeführt werden, den einen oder die mehreren Prozessoren veranlassen, Operationen auszuführen, die umfassen:
Zuordnen von Sichtprivilegien einer sicheren Sicht einer oder mehreren von mehreren zugrundeliegenden Einzelheiten eines Mitbenutzungsobjekts eines ersten Kontos, so dass jede der einen oder der mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts eine Definition der sicheren Sicht umfasst;
Gewähren einem zweiten Konto kontoübergreifende Zugriffsrechte auf das Mitbenutzungsobjekt des ersten Kontos;
in Reaktion auf das Empfangen einer Anforderung vom zweiten Konto, auf das Mitbenutzungsobjekt zuzugreifen, Modifizieren eines Abfrageplans der Anforderung unter Verwendung eines sicheren Projektionsoperators, um sicherzustellen, dass alle sicherheitsbezogenen Filterungen, die die Anforderung betreffen, stattfinden, bevor irgendein Ausdruck, der die Anforderung betrifft und sich außerhalb der sicheren Sicht befindet, ausgewertet wird, wobei die Ausdrücke, die nicht von der sicheren Sicht stammen, nicht unter eine Grenze der sicheren Sicht nach unten geschoben werden können, so dass die eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts vor dem zweiten Konto verborgen und für das erste Konto sichtbar sind; und
Zugreifen durch das zweite Konto auf das Mitbenutzungsobjekt, ohne eine Kopie von irgendeiner der einen oder der mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts im zweiten Konto zu erzeugen.

8. Computerprogramm nach Anspruch 7, wobei die Operationen ferner umfassen:
Erzeugen innerhalb des zweiten Kontos eines Aliasobjekts, das auf das Mitbenutzungsobjekt verweist; und
Verketten des Aliasobjekts mit der einen oder den mehreren von den mehreren der zugrundeliegenden Einzelheiten, wobei das zweite Konto die Anforderung, auf das eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten zuzugreifen, zu dem Aliasobjekt leitet und unter Verwendung der Zugriffsrechte des Mitbenutzungsobjekts auf das eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten zugreift.

9. Computerprogramm nach Anspruch 8, wobei das Aliasobjekt als ein Proxy für das Mitbenutzungsobjekt dient.

10. Computerprogramm nach Anspruch 7, wobei das Gewähren dem zweiten Konto kontenübergreifender Zugriffsrechte auf das Mitbenutzungsobjekt das Aufnehmen des zweiten Kontos in eine Liste von Konten, die Zugriffsrechte auf das Mitbenutzungsobjekt des ersten Kontos aufweisen, umfasst.

11. Computerprogramm nach Anspruch 7, das ferner umfasst:
Kommentieren jedes Ausdrucks der Anforderung von dem zweiten Konto mit den Sicherheitseigenschaften jedes Ausdrucks; und
für jeden Ausdruck, der als sicher bestimmt wird, Ermöglichen von Transformationen, die es ermöglichen, dass jeder Ausdruck durch eine Grenze einer sicheren Sicht geschoben wird, wobei ein Ausdruck als sicher bestimmt wird, falls bestimmt wird, dass er keine Fehler erzeugt und keine anwenderdefinierte Funktion (UDF) enthält.

12. Computerprogramm nach Anspruch 7, wobei eine zugrundeliegende Einzelheit des Mitbenutzungsobjekts eines oder mehrere des Folgenden umfasst: ein Datenfeld einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; ein Strukturelement einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; oder eine Datenmenge in dem Mitbenutzungsobjekt.

13. Nicht transitorisches computerlesbares Medium, das darin gespeicherte Anweisungen aufweist, die, wenn sie durch einen oder mehrere Prozessoren ausgeführt werden, den einen oder die mehreren Prozessoren veranlassen: Sichtprivilegien einer sicheren Sicht einer oder mehreren von mehreren zugrundeliegenden Einzelheiten eines Mitbenutzungsobjekts eines ersten Kontos zuzuordnen, so dass jede der einen oder der mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts eine Definition der sicheren Sicht umfasst; einem zweiten Konto kontoübergreifende Zugriffsrechte auf das Mitbenutzungsobjekt des ersten Kontos zu gewähren; in Reaktion auf das Empfangen einer Anforderung vom zweiten Konto, auf das Mitbenutzungsobjekt zuzugreifen, einen Abfrageplan der Anforderung unter Verwendung eines sicheren Projektionsoperators zu modifizieren, um sicherzustellen, dass alle sicherheitsbezogenen Filterungen, die die Anforderung betreffen, stattfinden, bevor irgendein Ausdruck, der die Anforderung betrifft und sich außerhalb der sicheren Sicht befindet, ausgewertet wird, wobei die Ausdrücke, die nicht von der sicheren Sicht stammen, nicht unter eine Grenze der sicheren Sicht nach unten geschoben werden können, so dass die eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts vor dem zweiten Konto verborgen und für das erste Konto sichtbar sind; und durch das zweite Konto auf das Mitbenutzungsobjekt zuzugreifen, ohne eine Kopie von irgendeiner der einen oder der mehreren von den mehreren der zugrundeliegenden Einzelheiten des Mitbenutzungsobjekts im zweiten Konto zu erzeugen.

14. Nicht transitorisches computerlesbares Medium nach Anspruch 13, wobei der eine oder die mehreren Prozessoren ferner: innerhalb des zweiten Kontos ein Aliasobjekt zu erzeugen, das auf das Mitbenutzungsobjekt verweist; und das Aliasobjekt mit der einen oder den mehreren von den mehreren der zugrundeliegenden Einzelheiten zu verketten, wobei das zweite Konto die Anforderung, auf das eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten zuzugreifen, zu dem Aliasobjekt leitet und unter Verwendung der Zugriffsrechte des Mitbenutzungsobjekts auf das eine oder die mehreren von den mehreren der zugrundeliegenden Einzelheiten zugreift.

15. Nicht transitorisches computerlesbares Medium nach Anspruch 14, wobei das Aliasobjekt als ein Proxy für das Mitbenutzungsobjekt dient.

16. Nicht transitorisches computerlesbares Medium nach Anspruch 13, wobei der eine oder die mehreren Prozessoren, um dem zweiten Konto kontenübergreifende Zugriffsrechte auf das Mitbenutzungsobjekt zu gewähren, das zweite Konto in eine Liste von Konten, die Zugriffsrechte auf das Mitbenutzungsobjekt des ersten Kontos aufweisen, aufnehmen sollen.

17. Nicht transitorisches computerlesbares Medium nach Anspruch 13, wobei der eine oder die mehreren Prozessoren ferner: jeden Ausdruck der Anforderung von dem zweiten Konto mit den Sicherheitseigenschaften jedes Ausdrucks zu kommentieren; und für jeden Ausdruck, der als sicher bestimmt wird, Transformationen zu ermöglichen, die es ermöglichen, dass jeder Ausdruck durch eine Grenze einer sicheren Sicht geschoben wird, wobei ein Aus-

druck als sicher bestimmt wird, falls bestimmt wird, dass er keine Fehler erzeugt und keine anwenderdefinierte Funktion (UDF) enthält.

18. Nicht transitorisches computerlesbares Medium nach Anspruch 13, wobei eine zugrundeliegende Einzelheit des Mitbenutzungsobjekts eines oder mehrere des Folgenden umfasst: ein Datenfeld einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; ein Strukturelement einer zugrundeliegenden Tabelle des Mitbenutzungsobjekts; oder eine Datenmenge in dem Mitbenutzungsobjekt.

Es folgen 16 Seiten Zeichnungen

Anhängende Zeichnungen

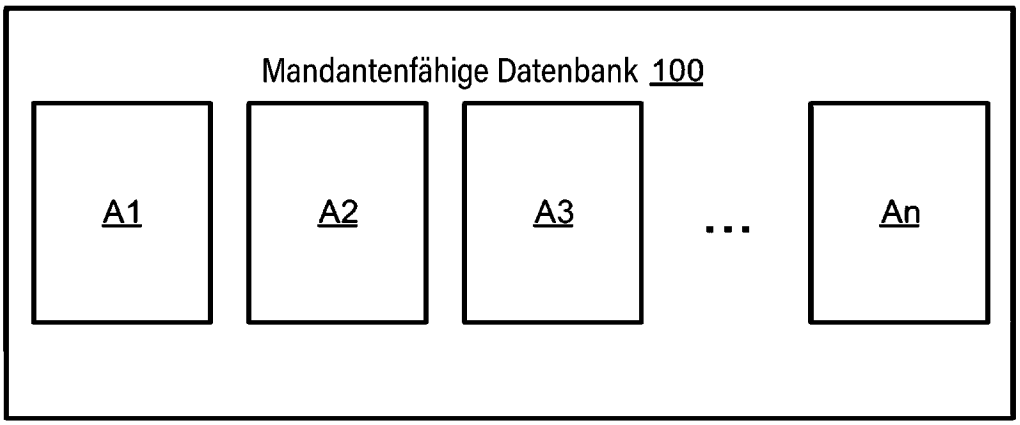


FIG. 1

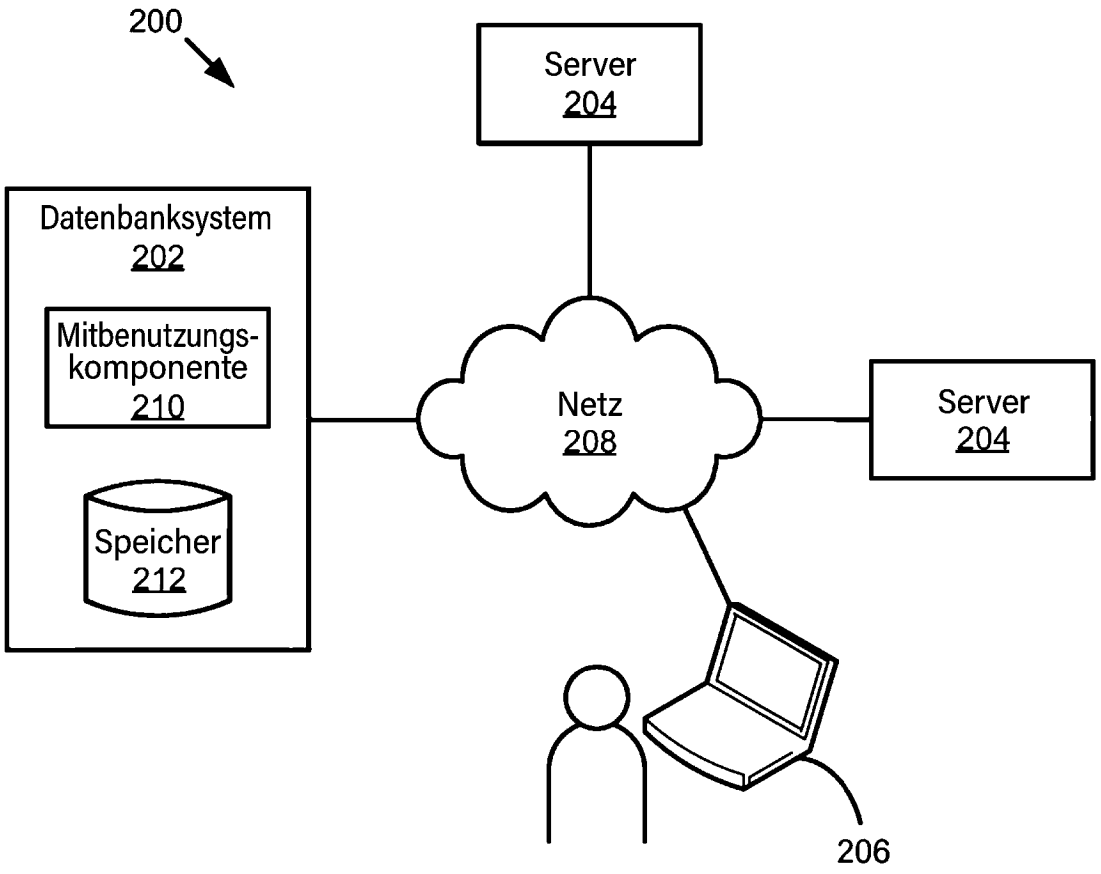


FIG. 2

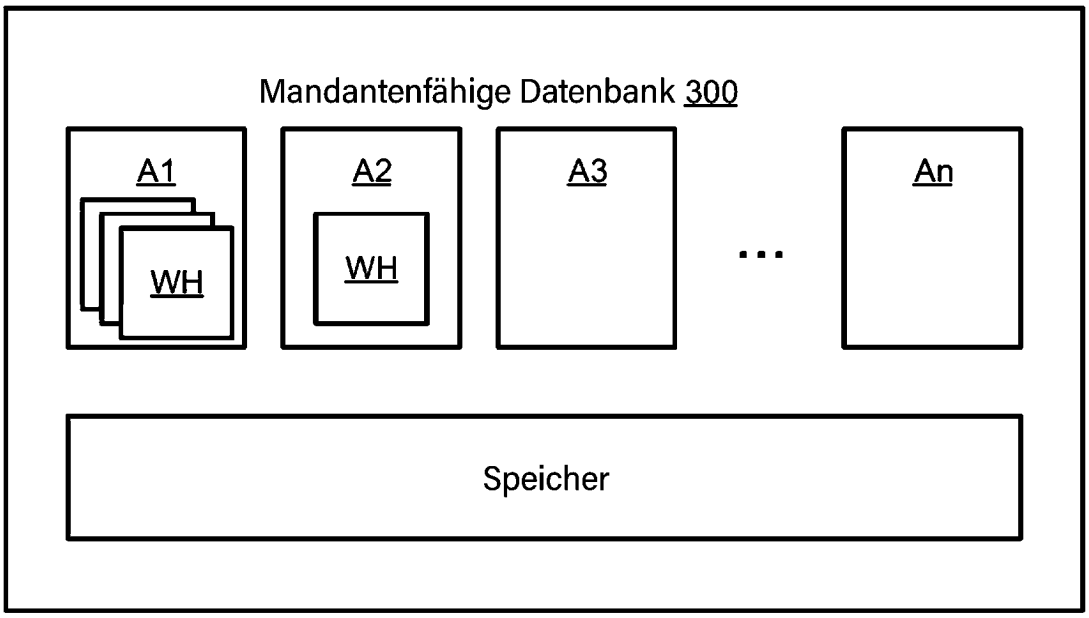

Mandantenfähige Datenbank 300
A1
WH
A2
WH
A3
...
An
Speicher

FIG. 3

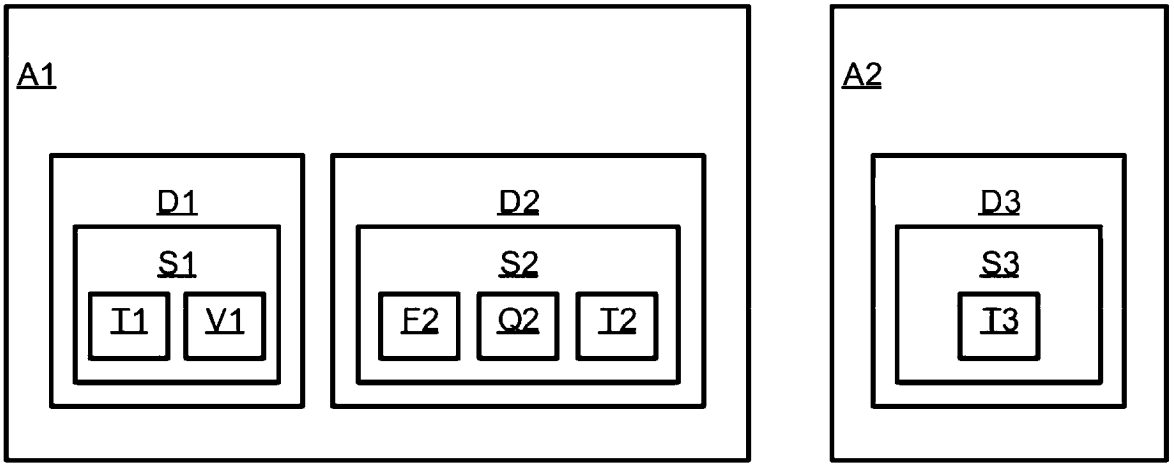

A1
D1
S1
T1
V1
D2
S2
F2
Q2
T2
A2
D3
S3
T3

FIG. 4

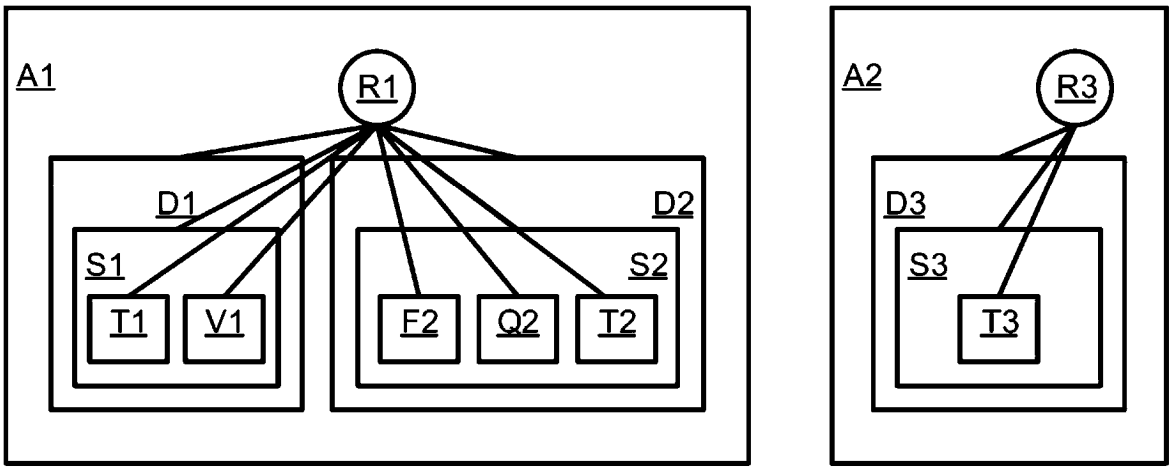

A1
R1
D1
D2
S1
S2
T1
V1
F2
Q2
T2
A2
R3
D3
S3
T3

FIG. 5

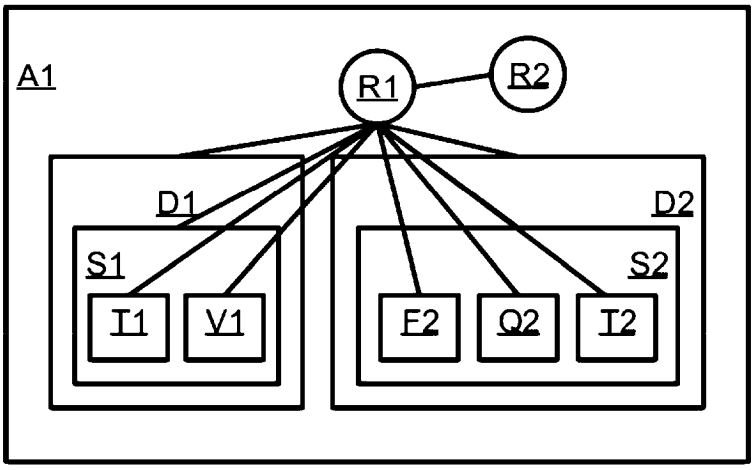

A1
R1
R2
D1
D2
S1
S2
T1
V1
F2
Q2
T2

FIG. 6

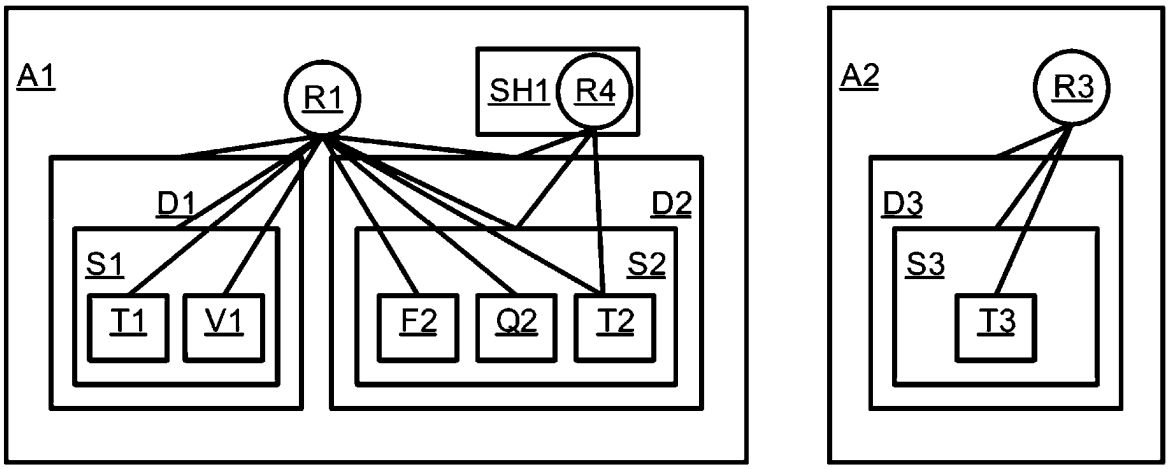

A1
R1
SH1
R4
D1
D2
S1
S2
T1
V1
F2
Q2
T2
A2
R3
D3
S3
T3

FIG. 7

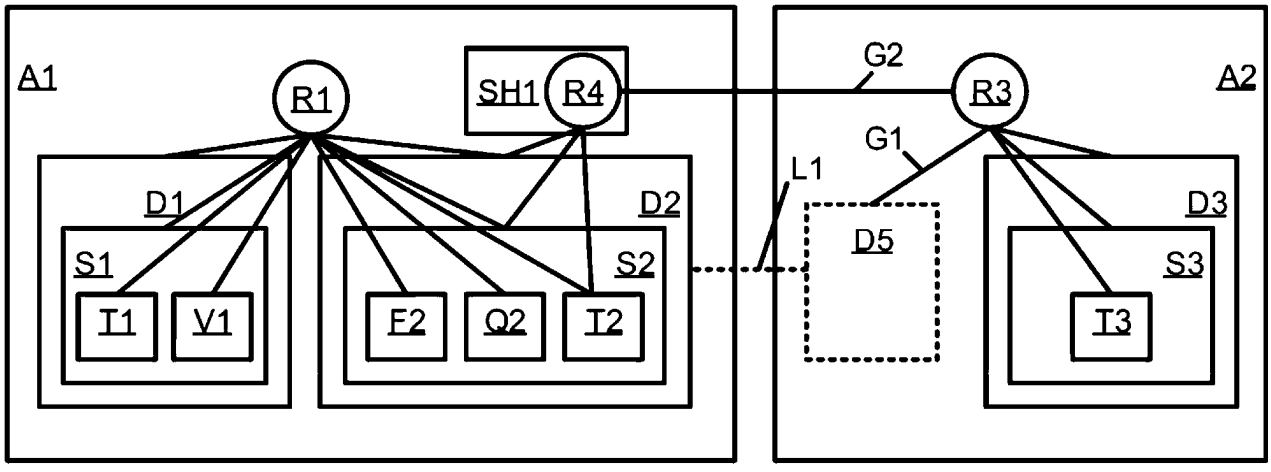

A1
R1
SH1
R4
G2
R3
A2
G1
L1
D1
D2
D5
D3
S1
S2
S3
T1
V1
F2
Q2
T2
T3

FIG. 8

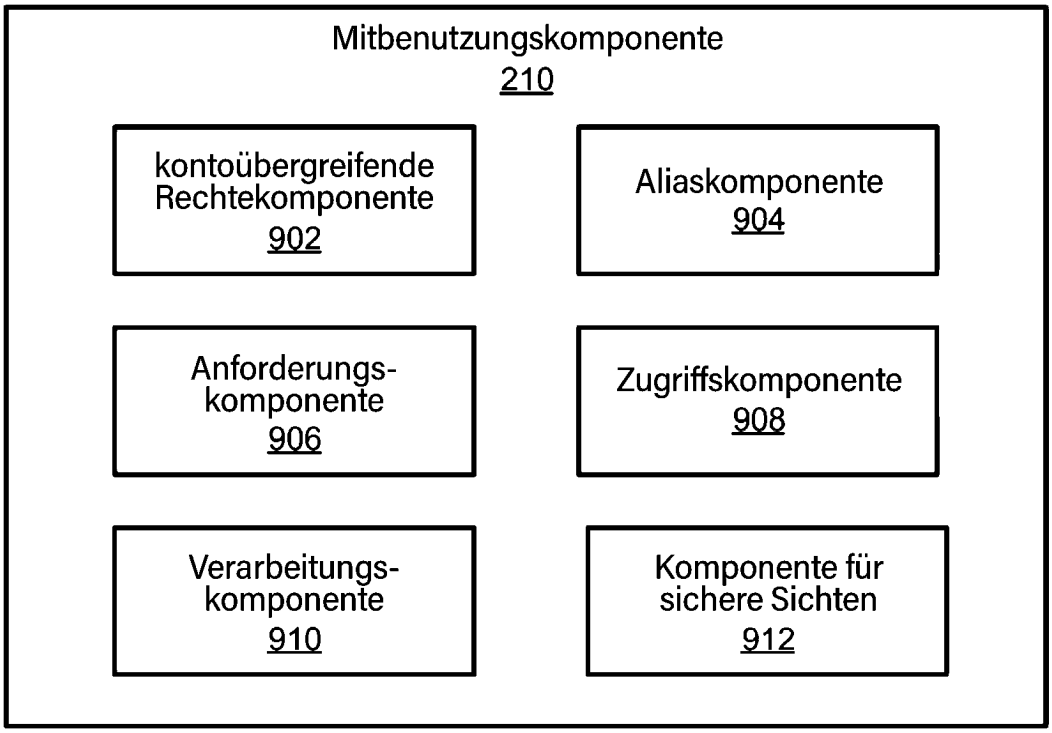

Mitbenutzungskomponente
210
kontoübergreifende Rechtekomponente
902
Aliaskomponente
904
Anforderungs-komponente
906
Zugriffskomponente
908
Verarbeitungs-komponente
910
Komponente für sichere Sichten
912

FIG. 9

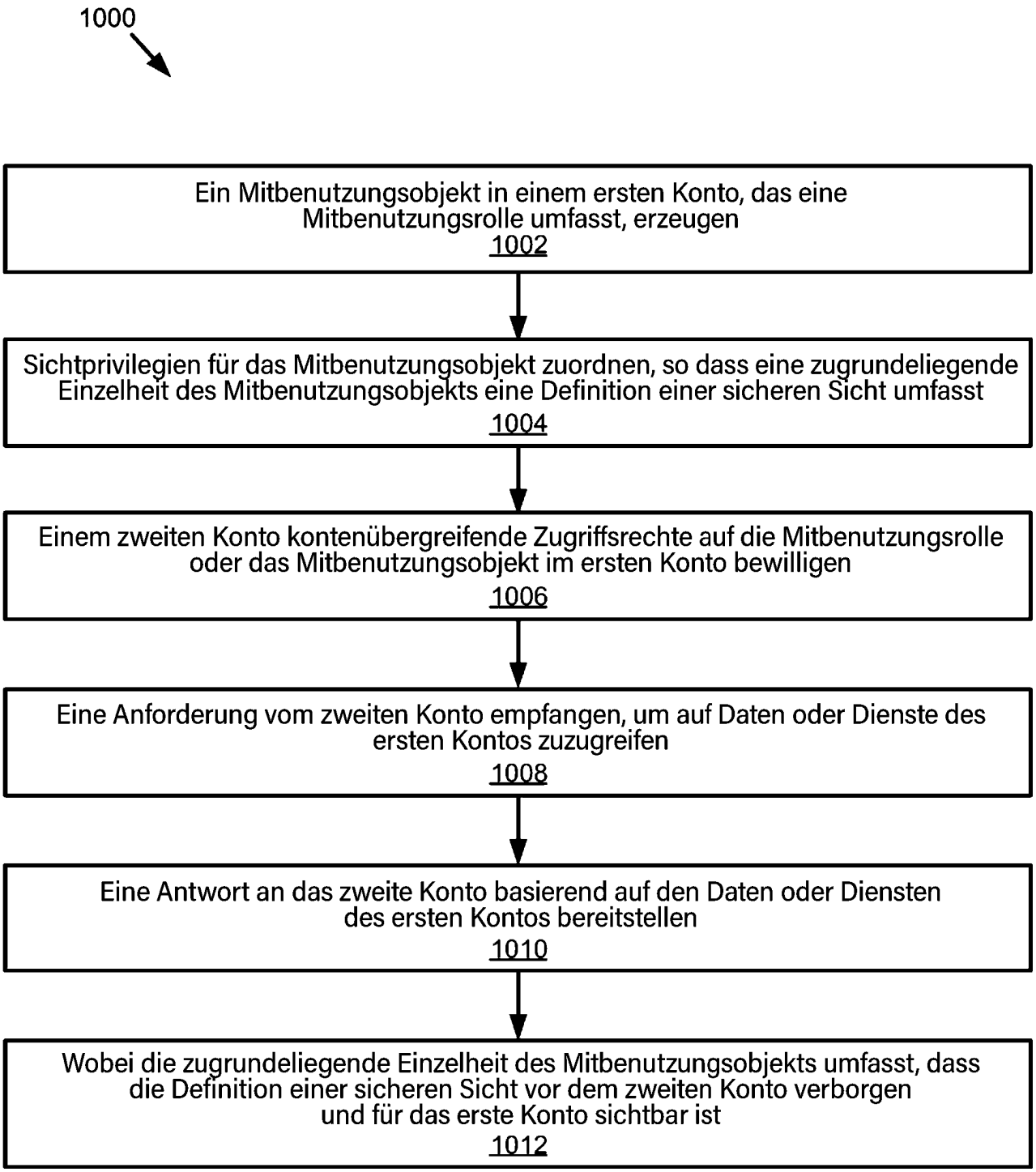

1000
Ein Mitbenutzungsobjekt in einem ersten Konto, das eine Mitbenutzungsrolle umfasst, erzeugen
1002
Sichtprivilegien für das Mitbenutzungsobjekt zuordnen, so dass eine zugrundeliegende Einzelheit des Mitbenutzungsobjekts eine Definition einer sicheren Sicht umfasst
1004
Einem zweiten Konto kontenübergreifende Zugriffsrechte auf die Mitbenutzungsrolle oder das Mitbenutzungsobjekt im ersten Konto bewilligen
1006
Eine Anforderung vom zweiten Konto empfangen, um auf Daten oder Dienste des ersten Kontos zuzugreifen
1008
Eine Antwort an das zweite Konto basierend auf den Daten oder Diensten des ersten Kontos bereitstellen
1010
Wobei die zugrundeliegende Einzelheit des Mitbenutzungsobjekts umfasst, dass die Definition einer sicheren Sicht vor dem zweiten Konto verborgen und für das erste Konto sichtbar ist
1012

FIG. 10

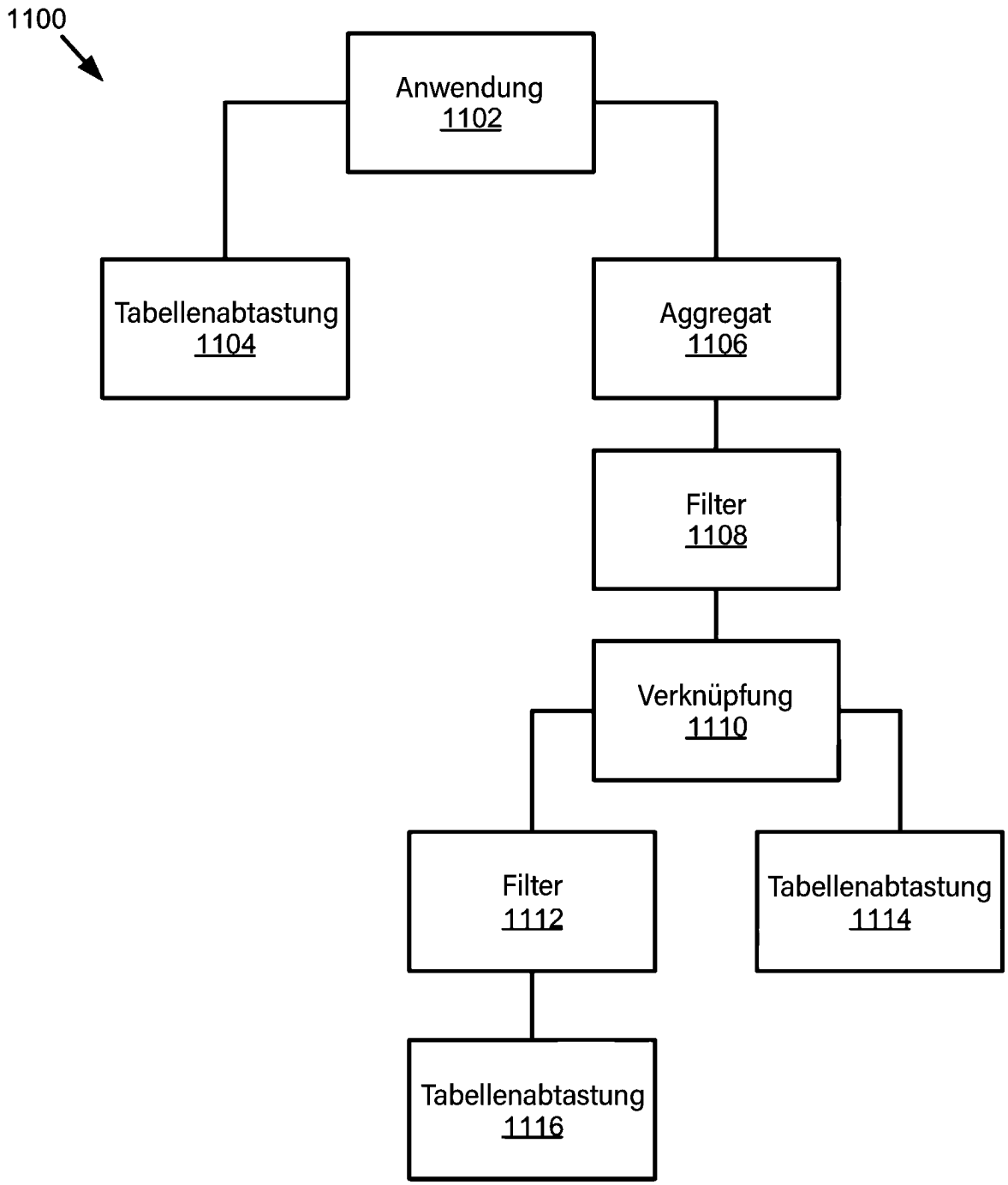

1100
Anwendung
1102
Tabellenabtastung
1104
Aggregat
1106
Filter
1108
Verknüpfung
1110
Filter
1112
Tabellenabtastung
1114
Tabellenabtastung
1116

FIG. 11

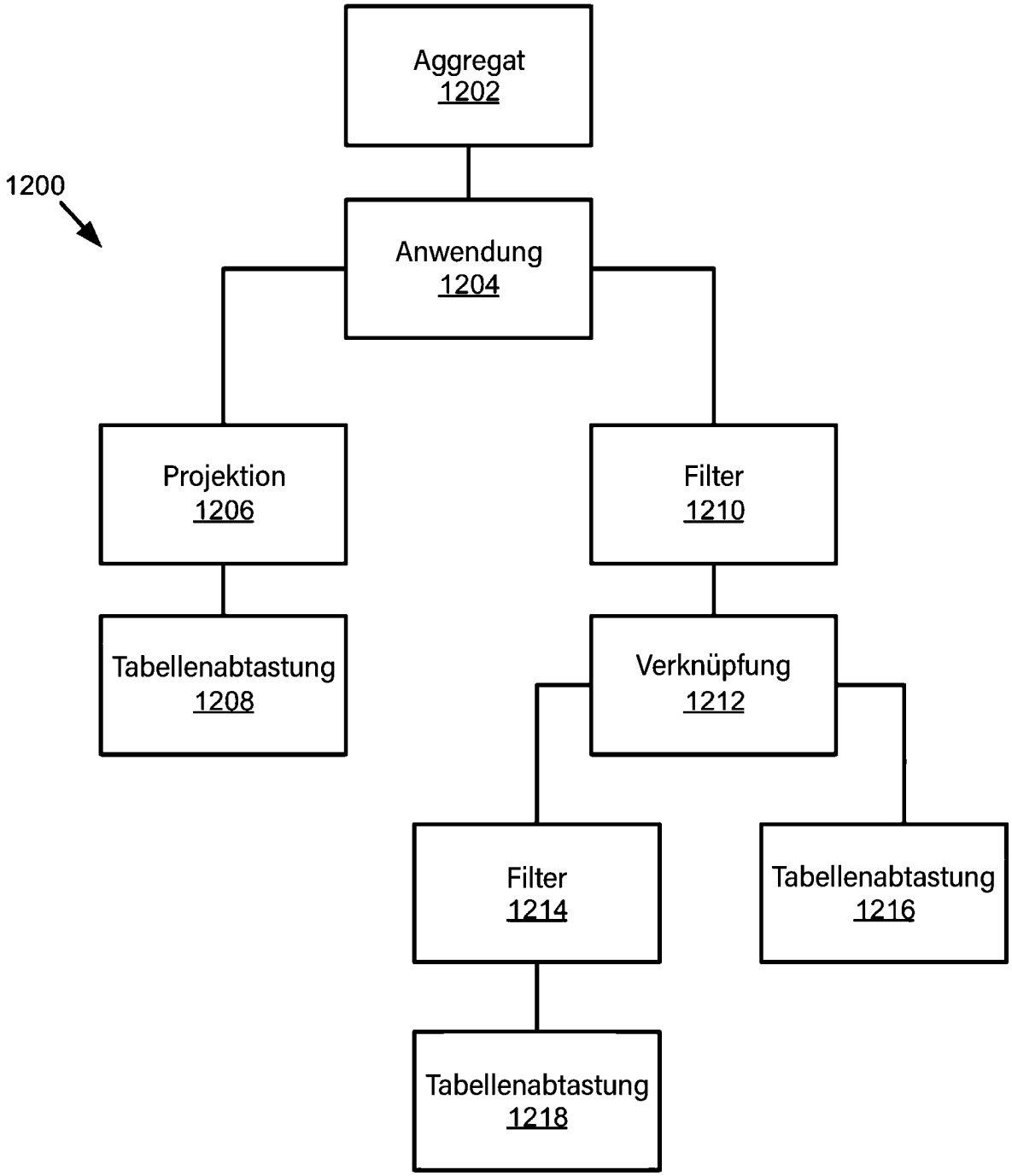

1200
Aggregat
1202
Anwendung
1204
Projektion
1206
Filter
1210
Tabellenabtastung
1208
Verknüpfung
1212
Filter
1214
Tabellenabtastung
1216
Tabellenabtastung
1218

FIG. 12

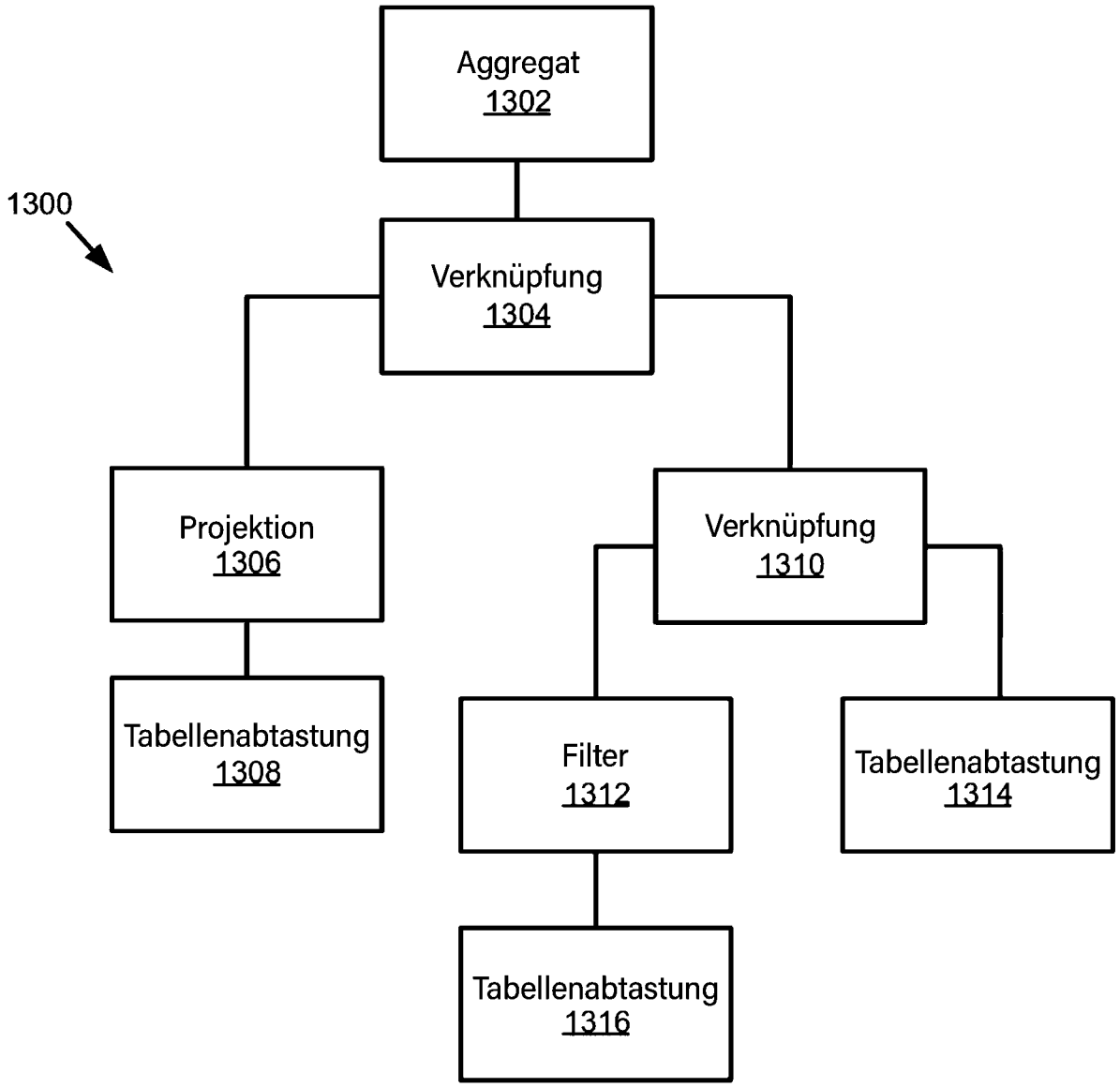

1300
Aggregat
1302
Verknüpfung
1304
Projektion
1306
Tabellenabtastung
1308
Verknüpfung
1310
Filter
1312
Tabellenabtastung
1316
Tabellenabtastung
1314

FIG. 13

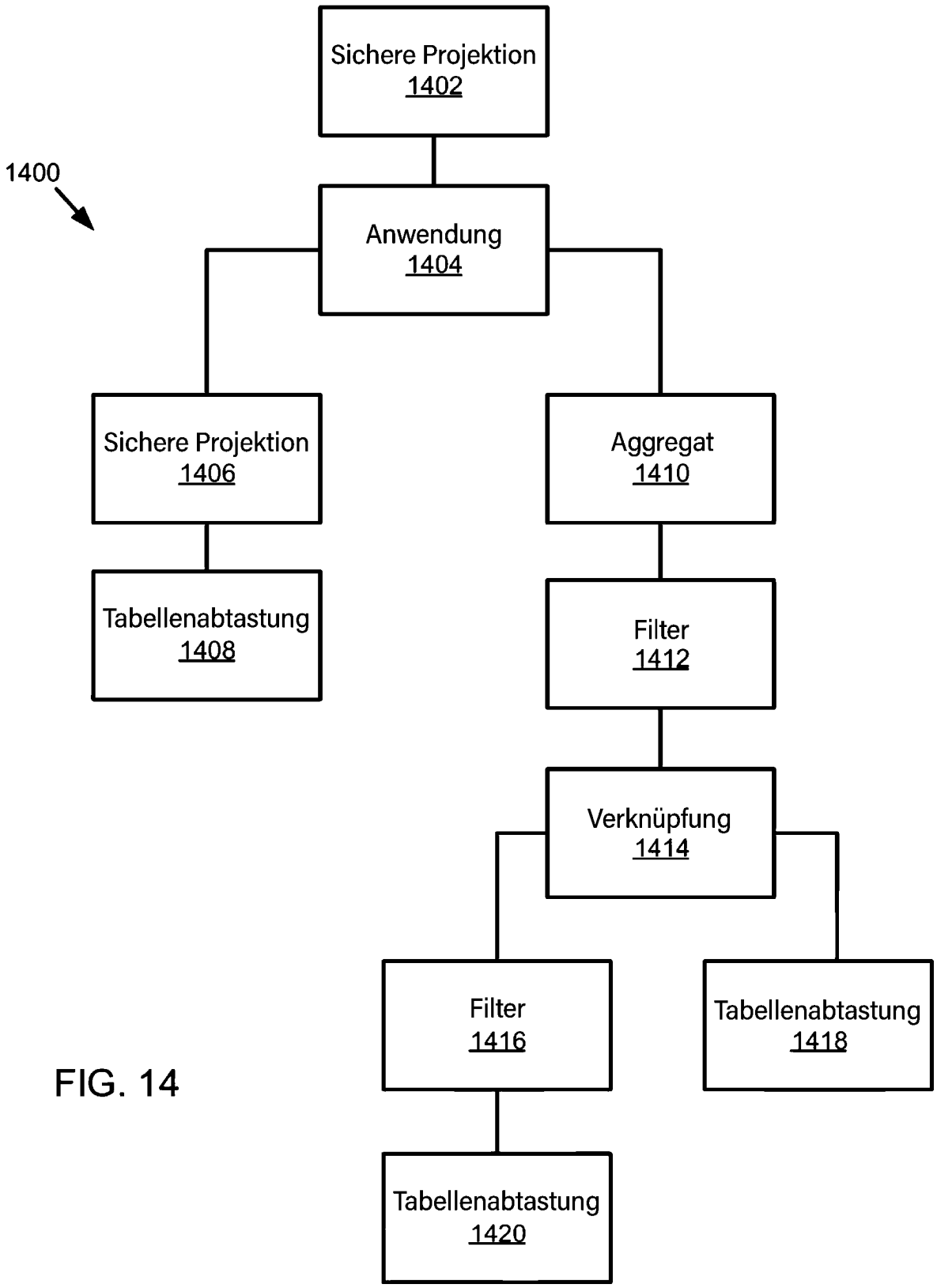

1400
Sichere Projektion
1402
Anwendung
1404
Sichere Projektion
1406
Tabellenabtastung
1408
Aggregat
1410
Filter
1412
Verknüpfung
1414
Filter
1416
Tabellenabtastung
1418
Tabellenabtastung
1420

FIG. 14

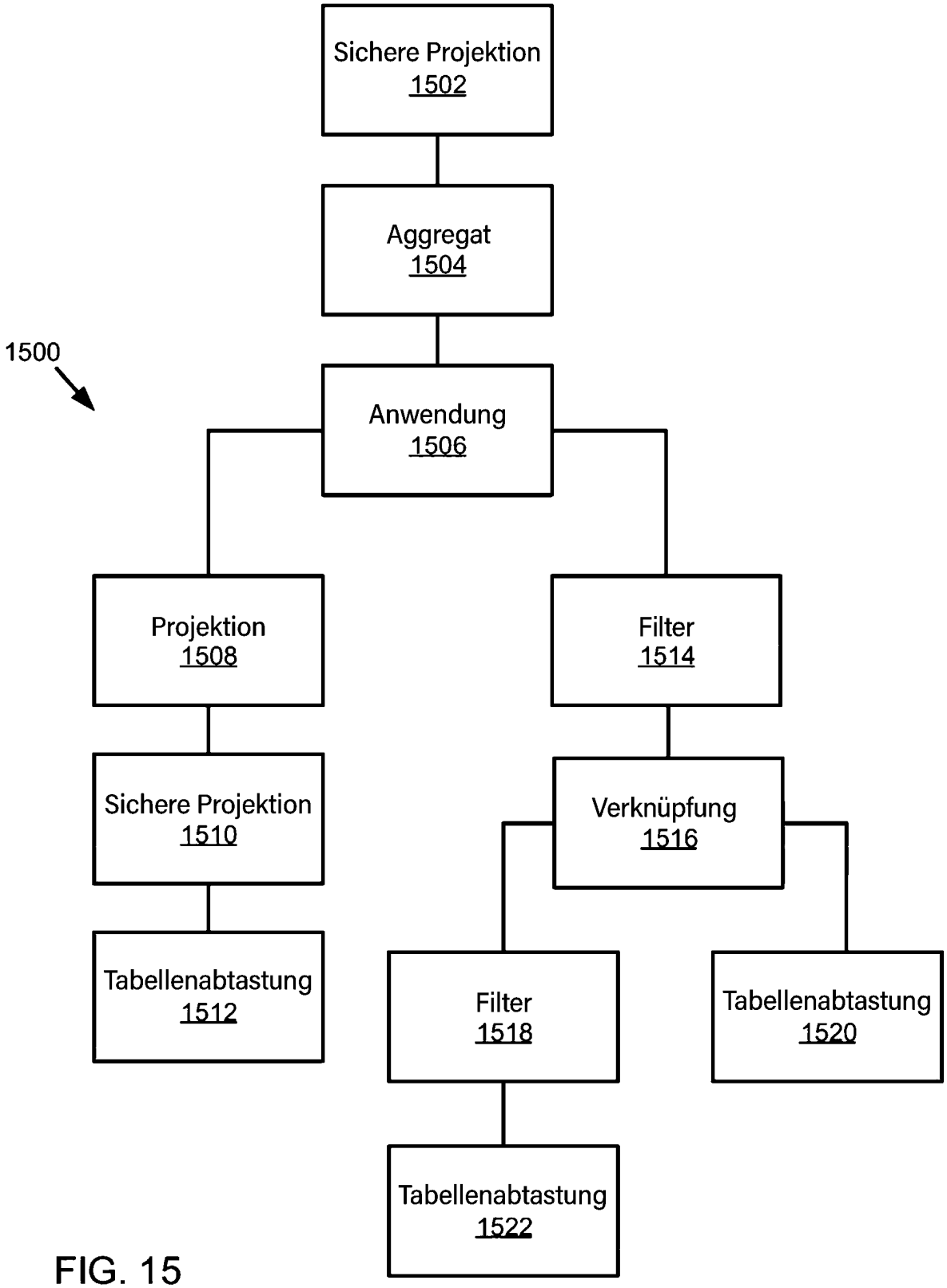

1500
Sichere Projektion
1502
Aggregat
1504
Anwendung
1506
Projektion
1508
Filter
1514
Sichere Projektion
1510
Verknüpfung
1516
Tabellenabtastung
1512
Filter
1518
Tabellenabtastung
1520
Tabellenabtastung
1522

FIG. 15

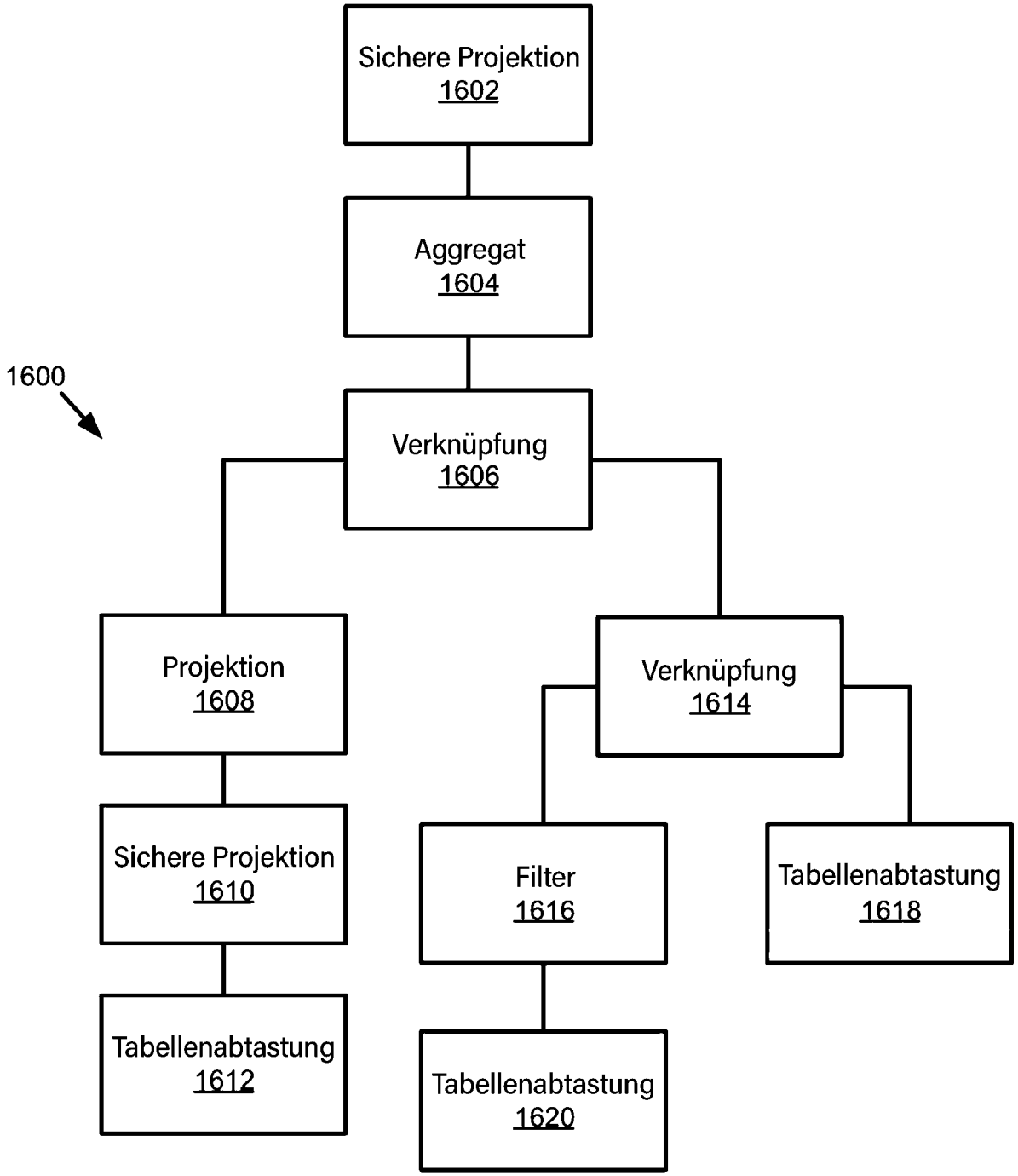

1600
Sichere Projektion
1602
Aggregat
1604
Verknüpfung
1606
Projektion
1608
Sichere Projektion
1610
Tabellenabtastung
1612
Verknüpfung
1614
Filter
1616
Tabellenabtastung
1620
Tabellenabtastung
1618

FIG. 16

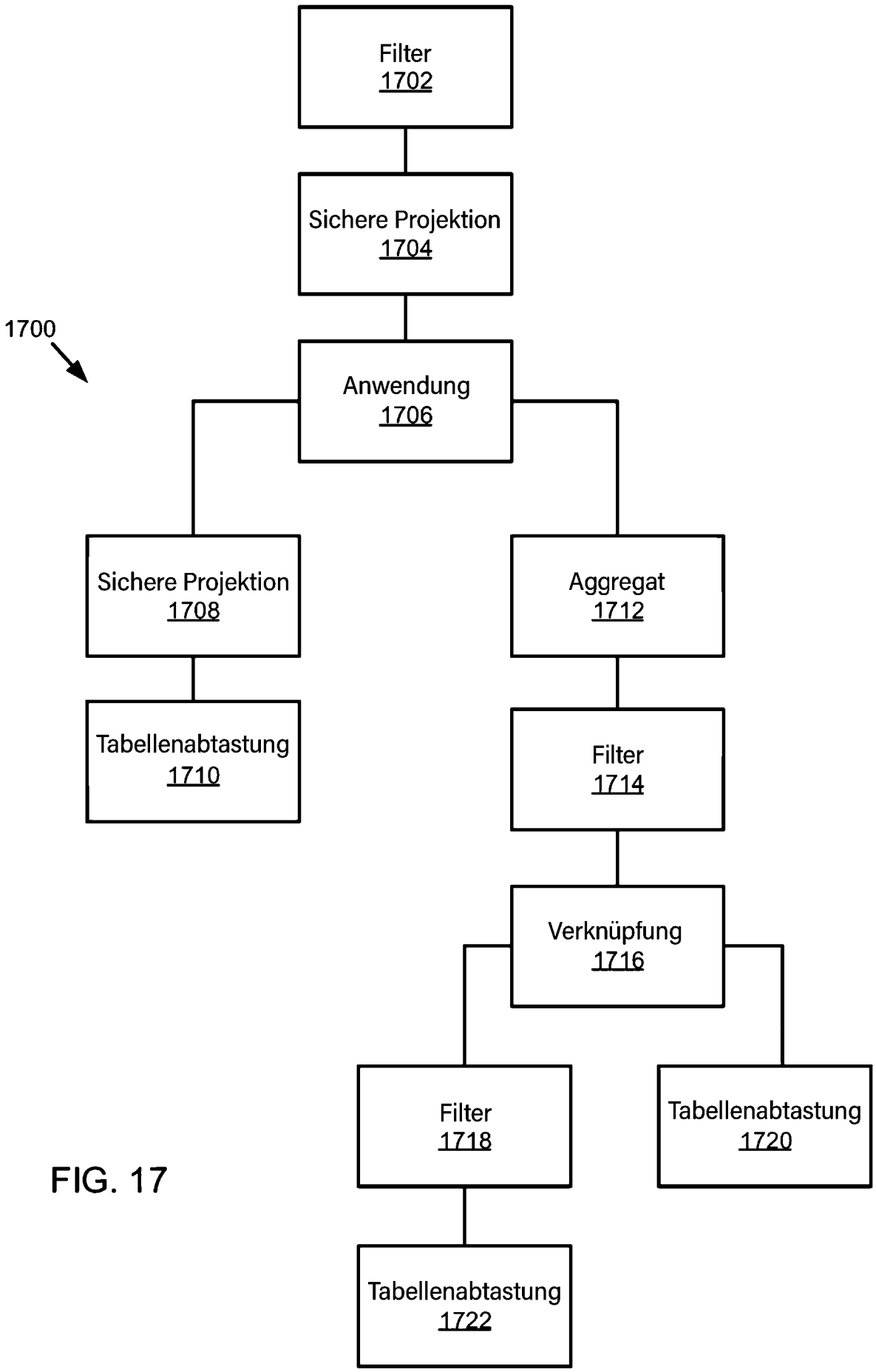

1700
Filter
1702
Sichere Projektion
1704
Anwendung
1706
Sichere Projektion
1708
Tabellenabtastung
1710
Aggregat
1712
Filter
1714
Verknüpfung
1716
Filter
1718
Tabellenabtastung
1720
Tabellenabtastung
1722

FIG. 17

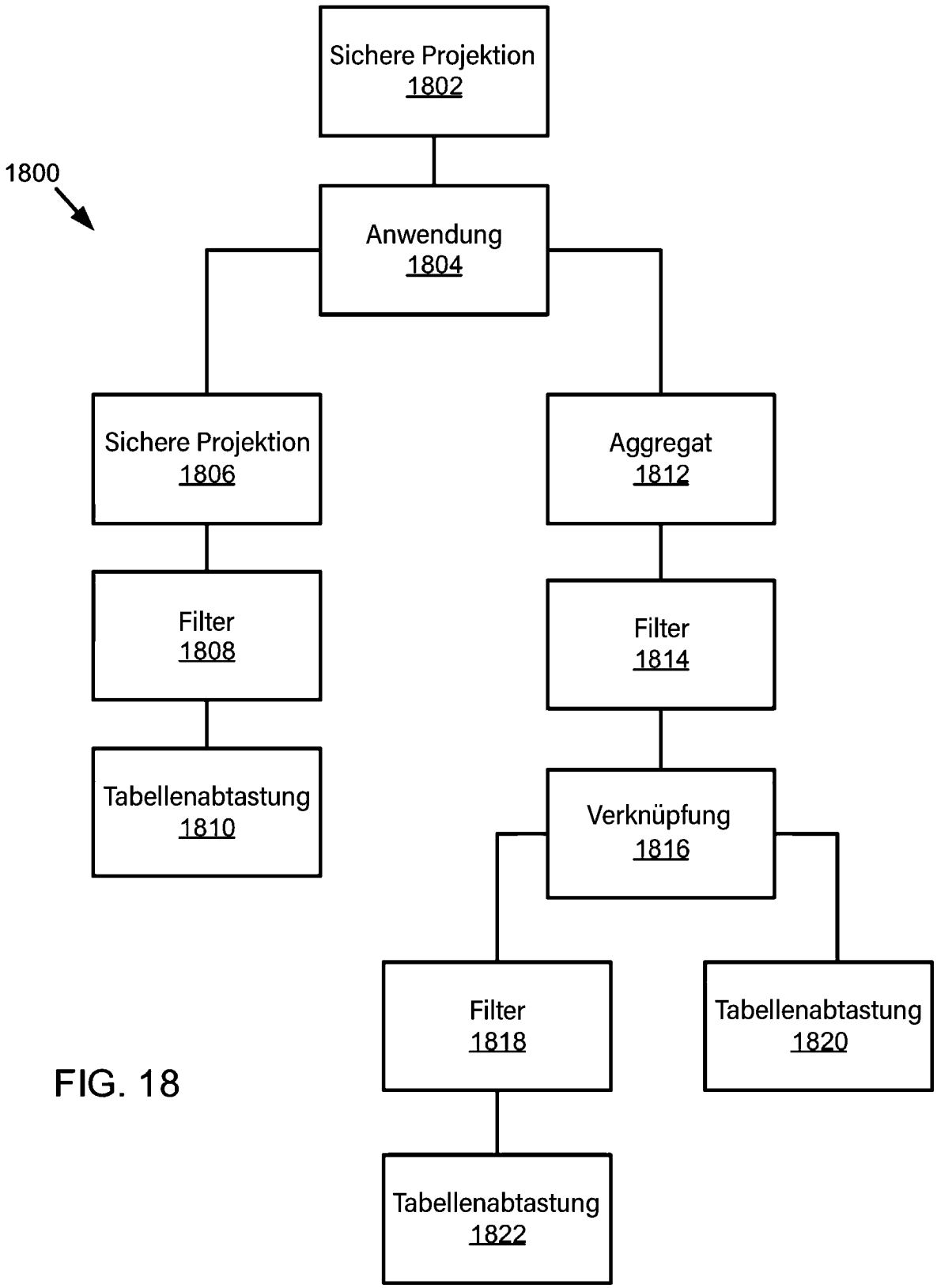

1800
Sichere Projektion
1802
Anwendung
1804
Sichere Projektion
1806
Filter
1808
Tabellenabtastung
1810
Aggregat
1812
Filter
1814
Verknüpfung
1816
Filter
1818
Tabellenabtastung
1820
Tabellenabtastung
1822

FIG. 18

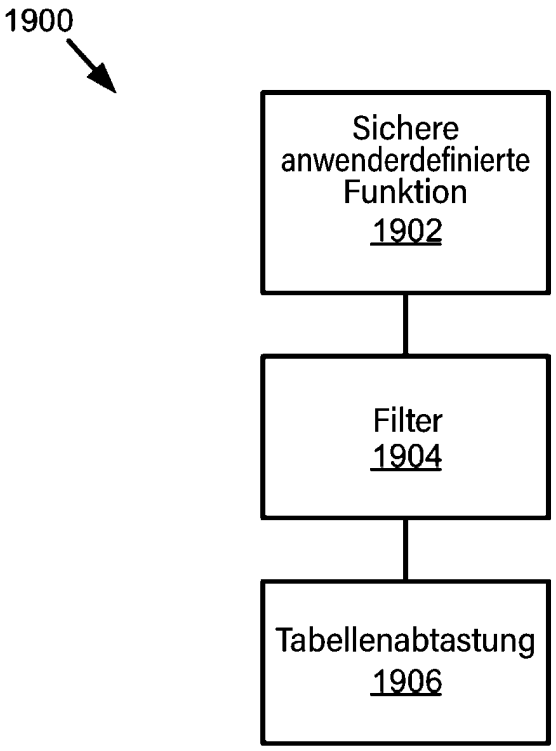

1900
Sichere anwenderdefinierte Funktion
1902
Filter
1904
Tabellenabtastung
1906

FIG. 19

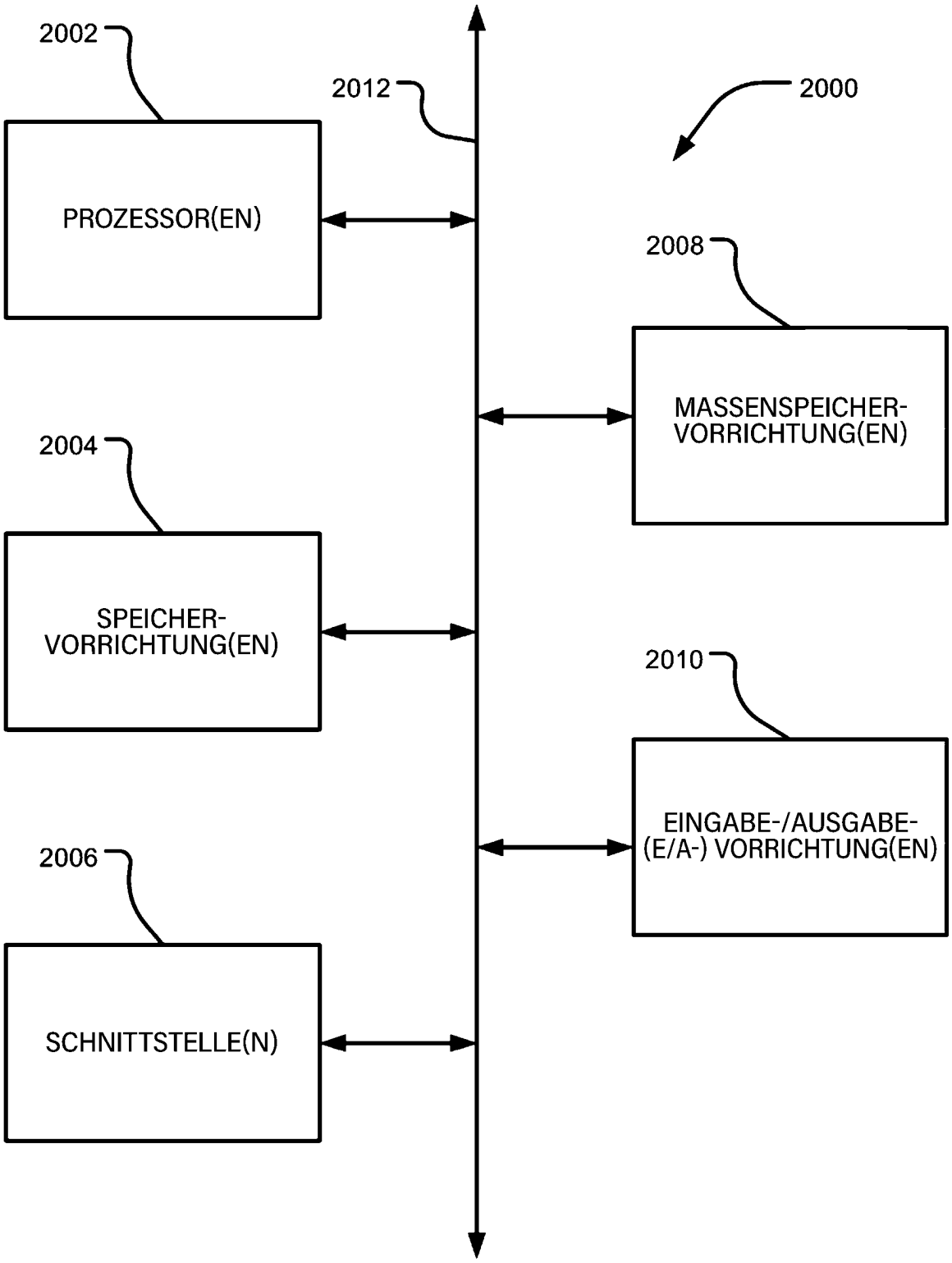


FIG. 20

Das Dokument wurde durch die Firma Jouve erstellt.