(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第6997774号
(P6997774)

(45)発行日　令和4年1月18日(2022.1.18)

(24)登録日　令和3年12月21日(2021.12.21)

(51)国際特許分類
Ｇ０６Ｆ　16/28　(2019.01)

ＦＩ
Ｇ０６Ｆ　16/28

請求項の数　20　（全40頁）

(21)出願番号　特願2019-517425(P2019-517425)
(86)(22)出願日　平成29年9月28日(2017.9.28)
(65)公表番号　特表2019-533245(P2019-533245A)
(43)公表日　令和1年11月14日(2019.11.14)
(86)国際出願番号　PCT/US2017/054090
(87)国際公開番号　WO2018/064375
(87)国際公開日　平成30年4月5日(2018.4.5)
審査請求日　令和2年9月24日(2020.9.24)
(31)優先権主張番号　15/283,119
(32)優先日　平成28年9月30日(2016.9.30)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　15/283,130
(32)優先日　平成28年9月30日(2016.9.30)
最終頁に続く

(73)特許権者　506332063
セールスフォース ドット コム インコーポレイティッド
アメリカ合衆国 カリフォルニア州 ９４１０５，サン フランシスコ， ミッション ストリート ４１５， サード フロアー
(74)代理人　100107766
弁理士　伊東 忠重
(74)代理人　100070150
弁理士　伊東 忠彦
(74)代理人　100091214
弁理士　大貫 進介
(72)発明者　ワーシャフスキー，アレックス
アメリカ合衆国 カリフォルニア州 ９４５９６，ウォールナット クリーク，フ
最終頁に続く

(54)【発明の名称】　マルチテナント非リレーショナル・プラットフォーム・オブジェクト

(57)【特許請求の範囲】
【請求項１】
データベース・システムであって
プロセッサと、
命令を記憶するメモリと、
を含み、
前記命令は、前記データベース・システムに、
複数のテナント企業のためのマルチテナント非リレーショナル・データベースを保持し、
前記マルチテナント非リレーショナル・データベースは、前記企業のための複数のレコードを記憶し、
前記企業のための前記レコードに関連する動的仮想テーブルを保持し、
前記データベース・システム内のカスタム・データ・オブジェクトを定義するための前記複数の企業のうち１つからのリクエストを処理し、前記リクエストは、リクエストしている企業を一意的に特定する少なくとも企業ＩＤを示し、前記カスタム・データ・オブジェクトの１つ以上の属性を特定し、
前記リクエストに基づいてカスタム・オブジェクト・スクリプトを生成し、前記カスタム・オブジェクト・スクリプトは、前記カスタム・データ・オブジェクト、一意的な企業ＩＤ、及び前記１つ以上の属性に対応する前記データベース・システムの１つ以上のデータベース列を定義し、
前記動的仮想テーブルを更新して、前記カスタム・オブジェクト・スクリプトによって定

義された前記１つ以上のデータベース列に対応する１つ以上の仮想列を含め、
前記マルチテナント非リレーショナル・データベースにおいて前記企業により共有されるテーブルの１つ以上の既存の列を更新して、前記１つ以上の仮想列と一致させ、
前記一意的な企業ＩＤに関連しない企業に対して、前記共有されるテーブルの前記１つ以上の既存の列へのアクセスを制限する
ことをさせるように構成可能である、データベース・システム。
【請求項２】
前記カスタム・オブジェクト・スクリプトは、前記一意的な企業ＩＤに関連付けられる、請求項１に記載のシステム。
【請求項３】
前記カスタム・データ・オブジェクトは、標準共有オブジェクトの拡張である、請求項１に記載のシステム。
【請求項４】
前記リクエストしている企業からの前記リクエストは、前記カスタム・データ・オブジェクトに関連する１つ以上のプライマリ・キー列を特定する、請求項１に記載のシステム。
【請求項５】
前記１つ以上のプライマリ・キー列は、前記一意的な企業ＩＤに対応する、請求項４に記載のシステム。
【請求項６】
前記命令は、前記データベース・システムに、
前記リクエストしている企業に対して、前記共有されるテーブルの前記１つ以上の既存の列へのアクセスを提供する、ことをさせるようにさらに構成可能である、請求項１に記載のシステム。
【請求項７】
前記命令は、前記データベース・システムに、
前記共有されるテーブルに１つ以上のレコードを追加し、前記１つ以上のレコードは、前記一意的な企業ＩＤ及び前記共有されるテーブルの前記１つ以上の既存の列に関連付けられる、ことをさせるようにさらに構成可能である、請求項１に記載のシステム。
【請求項８】
前記命令は、前記データベース・システムに、
前記共有されるテーブルの前記１つ以上の既存の列を１つ以上のプライバシー設定に関連付け、前記１つ以上のプライバシー設定は、前記１つ以上の既存の列の可視性を決定する、ことをさせるようにさらに構成可能である、請求項１に記載のシステム。
【請求項９】
前記１つ以上の属性は、前記企業によって定義されるカスタム属性である、請求項１に記載のシステム。
【請求項１０】
前記リクエストは、前記企業によって定義される相関ＩＤ属性を特定し、前記相関ＩＤ属性は、前記共有されるテーブルの１つ以上の列に関連するデータを集約するように構成可能である、請求項１に記載のシステム。
【請求項１１】
１つ以上のプロセッサにより実行される方法であって、
前記１つ以上のプロセッサにより、複数のテナント企業のためのマルチテナント非リレーショナル・データベースを保持することであって、前記マルチテナント非リレーショナル・データベースは、前記企業のための複数のレコードを記憶する、ことと、
前記１つ以上のプロセッサにより、前記企業のための前記レコードに関連する動的仮想テーブルを保持することと、
前記１つ以上のプロセッサにより、データベース・システム内のカスタム・データ・オブジェクトを定義するための前記複数の企業のうち１つからのリクエストを処理することであって、前記リクエストは、リクエストしている企業を一意的に特定する少なくとも企業

ＩＤを示し、前記カスタム・データ・オブジェクトの１つ以上の属性を特定する、ことと、
前記１つ以上のプロセッサにより、前記リクエストに基づいてカスタム・オブジェクト・スクリプトを生成することであって、前記カスタム・オブジェクト・スクリプトは、前記カスタム・データ・オブジェクト、一意的な企業ＩＤ、及び前記１つ以上の属性に対応する前記データベース・システムの１つ以上のデータベース列を定義する、ことと、
前記１つ以上のプロセッサにより、前記動的仮想テーブルを更新して、前記カスタム・オブジェクト・スクリプトによって定義された前記１つ以上のデータベース列に対応する１つ以上の仮想列を含めることと、
前記１つ以上のプロセッサにより、前記マルチテナント非リレーショナル・データベースにおいて前記企業により共有されるテーブルの１つ以上の既存の列を更新して、前記１つ以上の仮想列と一致させることと、
前記１つ以上のプロセッサにより、前記一意的な企業ＩＤに関連しない企業に対して、前記共有されるテーブルの前記１つ以上の既存の列へのアクセスを制限することと、
を含む方法。
【請求項１２】
前記カスタム・オブジェクト・スクリプトは、前記一意的な企業ＩＤに関連付けられる、請求項１１に記載の方法。
【請求項１３】
前記カスタム・データ・オブジェクトは、標準共有オブジェクトの拡張である、請求項１１に記載の方法。
【請求項１４】
前記１つ以上のプロセッサにより、前記共有されるテーブルに１つ以上のレコードを追加することであって、前記１つ以上のレコードは、前記一意的な企業ＩＤ及び前記共有されるテーブルの前記１つ以上の既存の列に関連付けられる、ことをさらに含む請求項１１に記載の方法。
【請求項１５】
前記１つ以上のプロセッサにより、前記共有されるテーブルの前記１つ以上の既存の列を１つ以上のプライバシー設定に関連付けることであって、前記１つ以上のプライバシー設定は、前記１つ以上の既存の列の可視性を決定する、ことをさらに含む請求項１１に記載の方法。
【請求項１６】
前記１つ以上の属性は、前記企業によって定義されるカスタム属性である、請求項１１に記載の方法。
【請求項１７】
前記リクエストは、前記企業によって定義される相関ＩＤ属性を特定し、前記相関ＩＤ属性は、前記共有されるテーブルの１つ以上の列に関連するデータを集約するように構成可能である、請求項１１に記載の方法。
【請求項１８】
１つ以上のプロセッサに動作を実行させるコンピュータ・プログラムであって、前記動作は、
複数のテナント企業のためのマルチテナント非リレーショナル・データベースを保持することであって、前記マルチテナント非リレーショナル・データベースは、前記企業のための複数のレコードを記憶する、ことと、
前記企業のための前記レコードに関連する動的仮想テーブルを保持することと、
データベース・システム内のカスタム・データ・オブジェクトを定義するための前記複数の企業のうち１つからのリクエストを処理することであって、前記リクエストは、リクエストしている企業を一意的に特定する少なくとも企業ＩＤを示し、前記カスタム・データ・オブジェクトの１つ以上の属性を特定する、ことと、
前記リクエストに基づいてカスタム・オブジェクト・スクリプトを生成することであって、前記カスタム・オブジェクト・スクリプトは、前記カスタム・データ・オブジェクト、

一意的な企業ＩＤ、及び前記１つ以上の属性に対応する前記データベース・システムの１つ以上のデータベース列を定義する、ことと、
前記動的仮想テーブルを更新して、前記カスタム・オブジェクト・スクリプトによって定義された前記１つ以上のデータベース列に対応する１つ以上の仮想列を含めることと、
前記マルチテナント非リレーショナル・データベースにおいて前記企業により共有されるテーブルの１つ以上の既存の列を更新して、前記１つ以上の仮想列と一致させることと、
前記一意的な企業ＩＤに関連しない企業に対して、前記共有されるテーブルの前記１つ以上の既存の列へのアクセスを制限することと、
を含む、コンピュータ・プログラム。
【請求項１９】
前記１つ以上の属性は、前記企業によって定義されるカスタム属性である、請求項１８に記載のコンピュータ・プログラム。
【請求項２０】
前記リクエストは、前記企業によって定義される相関ＩＤ属性を特定し、前記相関ＩＤ属性は、前記共有されるテーブルの１つ以上の列に関連するデータを集約するように構成可能である、請求項１８に記載のコンピュータ・プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本特許文書は、一般に、非リレーショナル・データベース・システムに関し、より詳細には、マルチテナント非リレーショナル・データベース・スキーマの更新及び管理に関する。
【０００２】
（著作権表示）
この特許文書の開示の一部は、著作権保護の対象となる資料を含んでいる。著作権所有者は、米国特許商標庁の特許ファイル又はレコードに現れる特許文書又は特許開示の複製に異議を申し立てないが、それ以外は何であろうとすべての著作権を留保する。
【０００３】
（優先権データ）
本特許文書は、以下の優先権の利益を主張する：
【０００４】
「マルチテナント非リレーショナル・プラットフォーム・オブジェクト（ＭＵＬＴＩ－ＴＥＮＡＮＴ　ＮＯＮ－ＲＥＬＡＴＩＯＮＡＬ　ＰＬＡＴＦＯＲＭ　ＯＢＪＥＣＴＳ）」と題する２０１６年９月３０日出願の米国特許出願第１５／２８３，１１９号（弁護士ドケット番号ＳＬＦＣＰ２２２／１７４９ＵＳ）；
【０００５】
「カスタム・マルチテナント非リレーショナル・プラットフォーム・オブジェクト（ＣＵＳＴＯＭ　ＭＵＬＴＩ－ＴＥＮＡＮＴ　ＮＯＮ－ＲＥＬＡＴＩＯＮＡＬ　ＰＬＡＴＦＯＲＭ　ＯＢＪＥＣＴＳ）」と題する２０１６年９月３０日に出願された米国特許出願第１５／２８３，１３０号（弁護士ドケット番号ＳＬＦＣＰ２２２Ａ／１７４９ＵＳ２）；
【０００６】
「マルチテナント非リレーショナル・プラットフォーム・オブジェクトのためのプロビジョニング（ＰＲＯＶＩＳＩＯＮＩＮＧ　ＦＯＲ　ＭＵＬＴＩ－ＴＥＮＡＮＴ　ＮＯＮ－ＲＥＬＡＴＩＯＮＡＬ　ＰＬＡＴＦＯＲＭ　ＯＢＪＥＣＴＳ）」と題する２０１６年９月３０日出願の米国特許出願第１５／２８３，１４５号（弁護士ドケット番号ＳＬＦＣＰ２２２Ｂ／１７４９ＵＳ３）
【０００７】
米国特許出願第１５／２８３，１１９号、米国特許出願第１５／２８３，１３０号及び米国特許出願第１５／２８３，１４５号の各々は、その全体及びすべての目的のために、参照により本明細書に組み込まれる。
【背景技術】

【０００８】
「クラウド・コンピューティング」サービスは、リクエストに応じて、共有リソース、アプリケーション、及び情報をコンピュータ及びその他のデバイスに提供する。クラウド・コンピューティング環境では、ソフトウェアを社内のコンピュータ・システムにローカルにインストールするのではなく、インターネット上でアクセス可能な１つ以上のサーバによってサービスを提供することができる。そのため、様々な役割を持つユーザは、クラウド・コンピューティング・サービスと対話することができる。
【図面の簡単な説明】
【０００９】
含まれる図面は、例示的な目的のためのものであり、マルチテナント非リレーショナル・データベース・スキーマを更新及び管理するための開示された発明のシステム、装置、方法、及びコンピュータ・プログラム製品の可能な構造及び動作の例を提供することにのみ役立つ。これらの図面は、開示された実施の精神及び範囲から逸脱することなく、当業者によってなされる形態及び詳細におけるいかなる変更も決して制限しない。
【００１０】
【図１】いくつかの実装による、マルチテナント非リレーショナル・データベース・スキーマを更新し、管理するためのシステム１００の一例のシステム図を示す。
【００１１】
【図２】いくつかの実装に従って実行される、マルチテナント非リレーショナル・データベース・システムにおけるデータベース・スキーマを更新及び管理するための方法２００の一例のフローチャートを示す。
【００１２】
【図３】この出願に記載される方法を使用する前に、非リレーショナル・データベースの物理テーブルに追加されるデータ・オブジェクトの例を示す。
【００１３】
【図４】いくつかの実装による、非リレーショナル共有テーブルに対する動的スキーマの例を示す。
【００１４】
【図５】いくつかの実装による、動的非リレーショナル共有テーブルに記憶されるいくつかのデータ・オブジェクト定義の例を示す。
【００１５】
【図６】いくつかの実装による、非リレーショナル・データベースにおける共有テーブルの物理レイアウトの例を示す。
【００１６】
【図７】いくつかの実装によって実行される、マルチテナント非リレーショナル・データベース・システムのためのカスタム・ベース・プラットフォーム・オブジェクトを作り出すための方法７００の一例のフローチャートを示す。
【００１７】
【図８】いくつかの実装によって実行される、アクセス制御を介して共有テーブル又はスキーマのサブセットを動的にプロビジョニングするための方法８００の一例のフローチャートを示す。
【００１８】
【図９Ａ】いくつかの実装によるオンデマンド・データベース・サービスを使用することができる環境１０の一例のブロック図を示す。
【００１９】
【図９Ｂ】図９Ａの要素のいくつかの実装及びこれらの要素間の種々の可能な相互接続の例のブロック図を示す。
【００２０】
【図１０Ａ】いくつかの実装による、オンデマンド・データベース・サービス環境９００のアーキテクチャ構成要素の一例のシステム図を示す。

【００２１】
【図１０Ｂ】いくつかの実装による、オンデマンド・データベース・サービス環境のアーキテクチャ構成要素の例をさらに例示するシステム図を示す。
【発明を実施するための形態】
【００２２】
開示された実装によるシステム、装置、方法、及びコンピュータ・プログラム製品の例は、この欄で説明される。これらの例は、単に文脈を追加し、開示された実装の理解を助けるために提供される。従って、これらの特定の詳細の一部又は全部を伴わずに実装を実施することができることは、当業者には明らかであろう。他の例では、不必要に不明瞭な実装を避けるために、特定の動作が詳細に記述されていない。他のアプリケーションも可能であり、以下の例を範囲又は設定のいずれにおいても決定的又は限定的なものとしてとらえるべきではない。
【００２３】
以下の詳細な説明では、明細書の一部を構成し、例示として具体的な実装を示す添付の図面を参照する。これらの実装は、当業者が開示された実装を実施することができるように十分に詳細に記載されているが、これらの例は限定的ではないことが理解され、他の実装を使用することができ、その精神及び範囲から逸脱することなく変更が行われ得ることが理解される。例えば、ここに示され、記載される方法の動作は、必ずしも示された順序で実行される必要はない。また、本方法は、示されたよりも多くの又は少ない動作を含んでもよいことも理解されたい。いくつかの実装では、別個の動作として本明細書で説明される動作を組み合わせることができる。逆に、本明細書において単一動作として記載されるものは、複数の動作において実施されてもよい。
【００２４】
開示されたシステム、装置、方法、及びコンピュータ・プログラム製品のいくつかの実装は、マルチテナント非リレーショナル・データベース・システムにおいてデータベース・スキーマを更新し管理するように構成される。
【００２５】
いくつかのマルチテナント・データベース・システムでは、顧客組織（すなわち、テナント）が１つの論理データベースでデータベース・リソースを共有するマルチテナント・アーキテクチャが使用される。データベース・テーブル自体は典型的には共有され、データ・モデルの各エンティティは、典型的には、各テナントの行を区別するｏｒｇａｎｉｚａｔｉｏｎ＿ｉｄ又はｔｅｎａｎｔ＿ｉｄ列を含む。テナントの文脈におけるクエリとデータ操作は、しばしば索引付けされるこのｔｅｎａｎｔ＿ｉｄ列を通してフィルターされ、適切なセキュリティとプライベート・データベースの外観を保証する。セールスフォース・ドットコム（ｓａｌｅｓｆｏｒｃｅ．ｃｏｍ）では、例えば、この戦略は、アカウント（Ａｃｃｏｕｎｔ）、コンタクト（Ｃｏｎｔａｃｔ：連絡先）、リード（Ｌｅａｄ）、機会（Ｏｐｐｏｒｔｕｎｉｔｙ）などの標準オブジェクトを顧客に公開するために使用される。
【００２６】
従来、標準オブジェクトは、オラクルなどのリレーショナル・データベースの共有テーブル内で使用されていた。このようなテーブルでは、標準オブジェクトは、１インスタンスあたり数万の顧客の間で共有することができる。新しいオブジェクトに対応する新しいフィールドを追加したり、フィールドを削除したりするなど、テーブルを根本的に変更しなければならない場合に、かなりの問題が発生する。リレーショナル・データベース・スキーマは、データベースの物理レベルで新しいテーブル、インデックスなどを定義するなどを含む、変更をする必要がある。データベースの所有者はいつでも自由に変更することはできず、むしろ、データベースは、特定の時間にのみロックダウンされ、変更されなければならず、その結果、非常に費用のかかる操作になる。
【００２７】
最近になって、このダウンタイム（ｄｏｗｎｔｉｍｅ）の除去が進んでいる。この方法で

は、新しいオブジェクトを含むフィールドを追加する必要がある場合に、新しい物理テーブルを定義するのではなく、行（ｒｏｗ）がメタデータ・テーブルに追加される。行は、仮想テーブルが別のテーブルにデータを動的に記憶していることを示す。物理データベース・レベルでは、単に行が挿入されているだけであり、これは低コストで非侵入的な操作であり、ダウンタイムを必要としない。次に、既存の列（ｃｏｌｕｍｎ）が既存のテーブルで更新され、新しいデータが追加される。従って、フィールドの追加は、リレーショナル・データベース・スキーマの変更の一部ではなく、アプリケーション・ロジックの一部である。

【００２８】

これは、リレーショナル・マルチテナント・データベースにとって非常に有用で費用効果が高いが、非リレーショナル・データベース・システムは、近年、非常に人気が高まっている。非リレーショナルデータベースシステムは、大量のデータに高速にアクセスする必要があるアプリケーションに対して理想的である。それらは、大きなデータセットのための柔軟でスケーラブルなデータベース・スキーマを提供する。このような非リレーショナル・データベースの１つは、ＨＢａｓｅである。非リレーショナル・データベースは、ログ内の大量のデータを高速でリアルタイムに取得することができる。従来のリレーショナル・データベースの厳密な構造ではなく、非リレーショナル・データベースは非構造化データベースを提供することができる。ＨＢａｓｅは、クラスタ上に分散される多くの列を指すポインタを使用してデータを記憶し、処理する列データベース（ｃｏｌｕｍｎ　ｄａｔａｂａｓｅｓ）を提供することができる。しかしながら、ＨＢａｓｅのような非リレーショナル・データベースは、リレーショナル・データベースとは異なって設計されるので、リレーショナル・データベースと同じ効率的で低コストな方法でオブジェクトとスキーマ管理を扱うフレームワークが欠けている。

【００２９】

例として、アクメ（Ａｃｍｅ）社は、数千の組織が大規模なデータセットをリアルタイムで取得するマルチテナント・データベース・システムを運営する会社である。Ａｃｍｅ社は、このデータの取得と記憶のためのＨＢａｓｅデータベースを保守する。ＨＢａｓｅ共有データベース「Ｌｏｇｉｎ＿Ｓｔａｔｕｓ」の列を、オブジェクト「Ｌｏｇｉｎ＿Ｅｖｅｎｔ」に関連付けて削除することにし、これは、ログイン・ステータスを別のフォーマットに変更したいからである。しかし、Ａｃｍｅは、単純にデータベース・スキーマを更新することはできないことを発見し、なぜなら、それはテーブルの操作を中断し、データベース・スキーマを更新することを伴い、大量のリアルタイム・データ収集に割り込むからである。Ａｃｍｅはまた、異なるオブジェクトとデータベース・フィールドを、顧客ごとにアクセス権を持つ特定の顧客にプロビジョニングする際に、いくつかの変更を加えたいと考えている。しかしながら、大規模な割り込みやスキーマの更新がなければ、これを行う方法はない。最後に、Ａｃｍｅは、組織が個々の組織のニーズに合わせて独自のオブジェクトやフィールドを作成できるようにしたいと考えているが、同じ理由でこれは困難であろう。

【００３０】

開示された技法のいくつかは、データベース構造を物理的に変更する必要はなく、アプリケーション層のスクリプトを介してマルチテナント・データベース・システム内のスキーマ要素を動的に更新するように実装することができる。新しい物理テーブルを定義する代わりに、データ・オブジェクトとしても知られるベース・プラットフォーム・オブジェクト（「ＢＰＯ（ｂａｓｅ　ｐｌａｔｆｏｒｍ　ｏｂｊｅｃｔ）」）が、ＸＭＬのようなスクリプト言語で定義される。ＢＰＯは、アカウントやリードのような標準オブジェクトのようなデータベース・オブジェクトで、１つ以上の組織のニーズのためにデータベースに素早く追加できる。動的仮想データベースは、共有スキーマを持つ仮想テーブルを含み、アプリケーション・レベルで保持される。ＢＰＯスクリプト定義は、仮想テーブルの列でＢＰＯを表すために使用される。次に、１つ以上の行が、新しいＢＰＯを表す物理的な非リレーショナル・テーブルに追加される。ＨＢａｓｅのような非リレーショナル・データベ

ースは、異なる行に対して様々な列の複数の構成が存在し得る列データベースを可能にするので、これは物理的な非リレーショナル列テーブルのプロパティに違反しない。従って、新しいオブジェクトがデータベース内で定義され、表現され、複数の組織がそのオブジェクトを利用してレコードを作成できる。これは、ゼロのダウンタイム又はデータベースの保存及び処理への割り込みで実行できる。

【００３１】

開示された技法のいくつかは、組織がマルチテナント非リレーショナル・データベース・システム内にカスタム・ベース・プラットフォーム・オブジェクト（「カスタムＢＰＯ（ｃｕｓｔｏｍ ＢＰＯｓ）」）を作成することを可能にするために実装することができる。データベース・システム内のすべての組織に供給される多数の標準オブジェクトを含む標準非リレーショナル・データベース・スキーマは、カスタム属性を持つ組織定義のカスタムＢＰＯで拡張できる。データベースのテナントは、ＸＭＬなどのスクリプトでオブジェクトを定義するか、共有スキーマから既存のオブジェクトを拡張する。このカスタム・オブジェクトとその属性は、動的仮想テーブルに含まれるように変換される。次に、物理的な非リレーショナル・テーブルが、カスタム・オブジェクトとその属性に対応する列で更新される。カスタム・オブジェクトは、それを作成したテナントに関連するデータベース内のレコードにのみ使用するように制限されており、この制限は、テナントのレコードに関連するテナント固有の識別子、すなわちｔｅｎａｎｔ＿ｉｄに基づいて行われる。

【００３２】

開示された技法のいくつかは、アクセス制御を介して、マルチテナント非リレーショナル・データベース・システム内の１つ以上のテナントに共有テーブルのサブセットへのアクセス権を動的にプロビジョニングするように実装することができる。標準スキーマは、非リレーショナルＢＰＯを使用して、共有テーブルのすべてのテナントで共有される。スキーマは、セグメントがアプリケーション層の特定のテナントに供給され、システム内の異なるＢＰＯセットへのアクセス権（ａｃｃｅｓｓ ｒｉｇｈｔ）とパーミッション権（ｐｅｒｍｉｓｓｉｏｎ ｒｉｇｈｔ）を非常に迅速に与えるように構成可能である。１つの組織は、そのデータ要件のために共有スキーマの特定の列にアクセスでき、一方、別の組織は、他の列にアクセスできる。

【００３３】

開示された技法のいくつかの実装を適用すると、上述のものに対する代替シナリオが提供される。この代替シナリオでは、Ａｃｍｅは再び、オブジェクト「Ｌｏｇｉｎ＿Ｅｖｅｎｔ」に関連するそのＨＢａｓｅ非リレーショナル共有データベースから列を削除することを決定した。新しい物理テーブルの定義を強制的に行う代わりに、データベースで発生するデータ収集と記憶の許容できない割り込みを招く代わりに、異なる設定（ｓｅｔｕｐ）を使用すると、より良いスキーマ管理が可能になる。この設定では、Ｌｏｇｉｎ＿Ｅｖｅｎｔに対する基本プラットフォーム・オブジェクト（「ＢＰＯ」）がＸＭＬスクリプトでＡｃｍｅによって定義される。動的仮想データベースは、共有スキーマを持つ仮想テーブルを含むＡｃｍｅによって保持される。Ｌｏｇｉｎ＿ＥｖｅｎｔのＢＰＯスクリプト定義は、アプリケーションを介して自動的にデータベース・オブジェクト定義、つまり「ビュー（ｖｉｅｗ）」に変換され、ＨＢａｓｅで読み込むことができる。このビューは、「Ｌｏｇｉｎ＿Ｓｔａｔｕｓ」列を含む、仮想テーブルの１つ以上の列のＬｏｇｉｎ＿Ｅｖｅｎｔを表すために使用される。「Ｌｏｇｉｎ＿Ｓｔａｔｕｓ」列を削除する場合は、アプリケーション層で仮想データベース内の列をシフトするスクリプトが書き込まれる。最初の列が消えると、スクリプトは仮想データベースに２番目の列からすべての列をコピーするように指示する。ＨＢａｓｅのデータベース層では、各列に明示的に定義されたデータ型が存在するのではなく、リレーショナル・データベースに存在するように、代わりに単純にバイトが存在し、任意の列がアプリケーション層で再定義されてもよい。従って、列を削除することは、列を下にシフトすることにより起こり、新しいテーブル全体を作成する必要はない。Ａｃｍｅが異なるデータ・タイプを持つＬｏｇｉｎ＿Ｓｔａｔｕｓ列を追加したい場合、変更されたＬｏｇｉｎ＿Ｓｔａｔｕｓで新しいＢＰＯを定義し、それを仮

想テーブルで使用するように変換し、次に、仮想テーブルの列をＨＢａｓｅテーブルにコピーすることで、簡単に追加できる。この場合も、ダウンタイムや新しいテーブルの作成は必要ない。

【００３４】

加えて、Ａｃｍｅのデータベースの任意のテナントは、自分自身のニーズに合わせて自身のカスタムＢＰＯを作成することができ、共有データベース内のそれらのＢＰＯに固有の唯一のアクセスを持つことができる。これは、スクリプトがｓｔａｎｄａｒｄ＿ｏｂｊｅｃｔの機能を拡張できるようにすることで達成できる。例えば、Ａｃｍｅの顧客が別のＬｏｇｉｎ＿Ｓｔａｔｕｓ形式の特別なＬｏｇｉｎ＿Ｅｖｅｎｔオブジェクトを望む場合、その顧客はそのオブジェクトを定義するＸＭＬスクリプトを書くことができ、そのスクリプトを変換してそのオブジェクトを仮想テーブルに入力することができる。すべての列にテナントごとにテーブルにおける所定のアクセス又は制限が可能であるため、仮想テーブルは、この顧客のみが自身のカスタムＢＰＯに関連した列を持つように構成可能である。オブジェクトは、そのレコードとカスタムＢＰＯに結び付けられた顧客の固有のｔｅｎａｎｔ＿ｉｄに基づいて、その顧客に対してのみアクセス可能な列の形式でＨＢａｓｅテーブルに移動することもできる。

【００３５】

Ａｃｍｅにとってのもう１つの利点は、これらの技法によって、種々のＢＰＯに対して様々な顧客にアクセス権を迅速に提供することができるということである。例えば、Ａｃｍｅは顧客のためにフリー・ティアとプレミアム・ティアを作成し、各々が共有テーブル内の異なるオブジェクトにアクセスできるようにしたい。ＢＰＯとカスタムＢＰＯがＨＢａｓｅテーブルに設置され、すべて顧客のｔｅｎａｎｔ＿ＩＤに結び付けられているため、アプリケーション層で仮想テーブルを介してデータベース・スキームを変更して、オブジェクトに関する異なるルールを宣言し、定義するのは簡単である。あるオブジェクトはプレミアム・ティアのみのアクセスを持つように定義されるべきであり、一方、他のオブジェクトはフリー・ティア及びプレミアム・ティアの両方のアクセスを持つように定義されるべきであると、Ａｃｍｅは容易に決めることができる。これは、ＸＭＬスクリプトで定義し、仮想テーブルに変換した後、ＨＢａｓｅ物理テーブルに移動して、フリー・ティアの顧客とプレミアム・ティアの顧客に対して異なる列を許可する。

【００３６】

すべてではないが、いくつかの実装では、開示された方法、装置、システム、及びコンピュータ読み取り可能な記憶媒体は、マルチテナント・データベース環境又はシステムにおいて使用するように構成又は設計されてもよい。

【００３７】

用語「マルチテナント・データベース・システム（ｍｕｌｔｉ－ｔｅｎａｎｔ　ｄａｔａｂａｓｅ　ｓｙｓｔｅｍ）」は、データベース・システムのハードウェア及びソフトウェアの様々な要素が１つ以上の顧客によって共有され得るシステムを指し得る。例えば、所定のアプリケーション・サーバは、多数の顧客に関するリクエストを同時に処理することができ、所定のデータベース・テーブルは、潜在的により多くの顧客のために、フィード項目のようなデータの行を保存することができる。用語「クエリ・プラン（ｑｕｅｒｙ　ｐｌａｎ：クエリ実行計画）」は、一般に、データベース・システム内の情報にアクセスするために使用される１つ以上の操作を指す。

【００３８】

図１は、いくつかの実装による、マルチテナント非リレーショナル・データベース・スキーマを更新し、管理するためのシステム１００の一例のシステム図を示す。システム１００は、互いに通信する様々な異なるハードウェア及び／又はソフトウェア構成要素を含む。図１の非限定的な例において、システム１００は、少なくとも１つの企業サーバ１０４、少なくとも１つのクライアント・システム１０８、少なくとも１つの非リレーショナル・データベース１１２、及び少なくとも１つの仮想データベース１１６を含む。

【００３９】

非リレーショナル・データベース１１２は、大量のデータセットの記憶及び読出しを可能にすることができる。非リレーショナル・データベース１１２は、ＨＢａｓｅ又は他の非リレーショナル・データベース管理システムに実装されたデータベースであり得る。このデータベースは、複数の企業（組織、又はテナントとも呼ばれる）のそれぞれについて１つ以上のレコードを含むことができる。いくつかの実装では、データベースは、複数の企業が同じテーブルにレコードを持ち、そのレコードのために同じ標準オブジェクトと列の多くを共有する、１つ以上の共有テーブルを含むことができる。いくつかの実装では、各企業は、非リレーショナル・データベース１１２内の特定の企業に固有の特定を提供するｔｅｎａｎｔ＿ｉｄに関連付けられる。例えば、エンティティＡｃｍｅは、レコード又はオブジェクトに関連するＡｃｍｅを一意的に特定する「１２３」のｔｅｎａｎｔ＿ｉｄを持ってもよい。共有テーブル内の他のテナントは、同じｔｅｎａｎｔ＿ｉｄを持つことはできない。

【００４０】

いくつかの実装では、非リレーショナル・データベース１１２は、分散された直線的にスケーラブルで一貫したキー値ストアの形態をとる１つ以上の共有テーブルを有する。キー値ストアでは、行内のデータが１つ以上の列によってグループ化される。列は、データベースに記憶されたデータの物理的な配置に影響する。列は、データベース・システム内の１つ以上のオブジェクトに基づいて定義される。すべての行に同じ列が含まれている必要はない。各行は、共有テーブル内の１つのレコードを表すことができ、行は、その行を一意的に特定する行キーを介してソートし及びクエリすることができる。行キーの一例は、共有テーブルのテナントを一意的に特定するｔｅｎａｎｔ＿ｉｄである。

【００４１】

いくつかの実装では、非リレーショナル・データベース１１２は、リレーショナル・データベースの機能性を非リレーショナル・データベース１１２に提供する１つ以上のアプリケーションと連携して動作してもよい。例えば、リレーショナル・データベース、構造化スキーマ、データ・タイプ、及びＳＱＬクエリの外観を提供することができる。そのようなアプリケーションの一例はフェニックス（Ｐｈｏｅｎｉｘ）であり、これはＨＢａｓｅ及び１つ以上のドライバと連携して動作し、ＨＢａｓｅ非リレーショナル・データベースにリレーショナル特徴を提供する。

【００４２】

仮想データベース１１６は、システム１００のアプリケーション・レベルに存在するデータベースである。いくつかの実装では、仮想データベース１１６は、１つ以上のソフトウェア・アプリケーションの内部で、又はそれと共に実行されてもよい。仮想データベース１１６は、データが物理的又は低レベルのデータベースに記憶されないという点で非リレーショナル・データベース１１２とは異なる。代わりに、データは、半構造化ソース及び典型的なリレーショナル又は非リレーショナル・データベース記憶方法を除いた他の方法を介して、アプリケーション層又はローカル又はリモート記憶装置に仮想的に記憶することができる。仮想データベース１１６は、従来のデータベースの低レベルでデータを記憶しないので、スキーマ管理及び修正に関しては、制限されない。仮想データベースの構造は、アプリケーション層ですばやく変更できる。

【００４３】

企業サーバ１０４は、システム１００の他の構成要素と通信することができる。この通信は、ネットワークとインターフェースの組み合わせを通して容易にすることができる。企業サーバ１０４は、クライアント・システム１０８からのデータ・リクエストに対処し、処理することができる。同様に、企業サーバ１０４は、データ・リクエストが処理された後に、クライアント・システム１０８に応答を返すことができる。例えば、企業サーバ１０４は、非リレーショナル・データベース１１２又は仮想データベース１１６などの１つ以上のデータベースからデータを読み出すことができる。それは、異なるデータベースからのデータの一部又は全部を組み合わせ、処理されたデータをクライアント・システム１０８に送ることができる。

【００４４】
クライアント・システム１０８は、１つ以上のデータ・ネットワークを介してサーバと通信することができるコンピューティング・デバイスであってもよい。クライアント・システム１０８の例は、デスクトップ・コンピュータ又はスマートフォン、タブレット、ラップトップ、Ｇｏｏｇｌｅ　Ｇｌａｓｓ（登録商標）などのウェアラブル・デバイス、他の光学ヘッドマウントディスプレイデバイス（ＯＨＭＤ）、スマートウォッチなどのポータブル電子デバイスが挙げられる。クライアント・システム１０８は、アプリケーションが展開され得る少なくとも１つのブラウザを含む。
【００４５】
図２は、いくつかの実装に従って実行される、マルチテナント非リレーショナル・データベース・システムにおけるデータベース・スキーマを更新し、管理するための方法２００の一例のフローチャートを示す。本明細書中に記載される方法２００及び他の方法は、図１のシステム１００を用いて実施することができるが、そのような方法の実装は、システム１００に限定されない。
【００４６】
ブロック２１０において、システム１００は、各々が複数のレコードを有する、複数の企業と関連するマルチテナント非リレーショナル・データベース１１２を保持する。いくつかの実装では、複数企業は、システム１００の各ユーザであり、レコードの形式でデータを記憶し処理することができる。レコードは、非リレーショナル・データベース１１２の共有テーブルの一部であってもよい。いくつかの実装では、各レコードは、オブジェクトを表す多数の列と共有テーブルの行の形式をとる。いくつかの実装では、列の数、種類、及びサイズは、レコードに関連する企業及びその企業のデータ・オブジェクトによって変化し得る。標準オブジェクトの場合は、標準オブジェクトの属性を示す列が、すべての企業に対して、又は共有テーブルの企業から指定されたパーミッション・セットとして現れることがある。例えば、標準オブジェクト「Ｕｓｅｒ＿Ｐｒｏｆｉｌｅ」は、Ｕｓｅｒ＿Ｐｒｏｆｉｌｅオブジェクトに関連する属性Ｕｓｅｒｎａｍｅ、Ｕｓｅｒ＿Ａｇｅ、及びＵｓｅｒ＿Ｌｏｃａｔｉｏｎと共有テーブルのすべての企業によってアクセスされるように指定されてもよい。これらの属性のそれぞれには、各企業の各レコードに表示される共有テーブルの列がある。いくつかの実装では、カスタム・オブジェクトは、限られた一連の企業に対して指定されることがある。例えば、ＡｃｍｅがＡｃｍｅ＿Ｕｓｅｒのカスタム・オブジェクトをその目的のために特別に作成している場合、その次に、Ａｃｍｅのレコードのみがａｃｍｅ＿Ｕｓｅｒオブジェクト及び関連する列をそのレコードに含めることができる。従って、一部の企業は他の企業とは異なる列にアクセスでき、一部のレコード（従って行）は他のレコードとは異なる列を含むことがある。
【００４７】
いくつかの実装では、マルチテナント非リレーショナル・データベース１１２の各テナント又は企業は、企業を一意的に特定する企業識別子（企業ＩＤ（ｅｎｔｅｒｐｒｉｓｅＩＤ））と関連付けられる。いくつかの実装では、企業特定は、英数字の一意的な数又は文字列であってもよい。いくつかの実装では、非リレーショナル・データベース１１２内の共有テーブルの各行（及びレコード）は、企業ＩＤの列を有し、これは、例えば、「ｔｅｎａｎｔ＿ｉｄ」、「ｅｎｔｅｒｐｒｉｓｅ＿ｉｄ」又は「ｏｒｇ＿ｉｄ」と名前をつけることができる。この企業ＩＤ列は、テーブルの行キーとして指定できる。共有テーブルのレコードは、ｅｎｔｅｒｐｒｉｓｅ＿ｉｄ行キーによってソートされ、ｅｎｔｅｒｐｒｉｓｅ＿ｉｄに基づいてクエリすることができる。このようにして、各レコードは、そのレコードに関連する企業に基づいて、容易にソート、検索、及び読み出される。
【００４８】
ブロック２２０において、システムは、マルチテナント非リレーショナル・データベース１１２のレコードに関連する動的仮想テーブルを保持する。動的仮想テーブルは、システム１００の仮想データベース１１６の一部であってもよい。いくつかの実装では、動的仮想テーブルは、システム１００内のアプリケーションの一部、又はアプリケーションと連

動する機能である。いくつかの実装において、マルチテナント非リレーショナル・データベース１１２に記憶されたすべてのレコードのサブセットは、動的仮想テーブルに記憶されてもよい。いくつかの実装では、上述の企業ＩＤは、ソーティング及びクエリするための各仮想テーブルのレコードの行キーとして指定されてもよい。

【００４９】

ブロック２３０において、システムは、非リレーショナル・データベース１１２のユーザから、データベース内のデータ・オブジェクトを定義するためのリクエストを受信する。リクエストは、データ・オブジェクトの少なくとも１つ以上の属性を特定する。いくつかの実装では、ユーザからのリクエストはクライアント・システム１０８から来る。いくつかの実装では、リクエストは企業サーバ１０４から来る。ユーザは、企業又は企業の代表的なメンバー、システム１００の開発者又は保守者、マルチテナント非リレーショナル・データベース１１２の開発者又は保守者、又は他のユーザであってもよい。いくつかの実装では、リクエストは宣言型言語の１つ以上のドキュメントの形式をとる。例えば、リクエストはＸＭＬ又はＪＳＯＮファイルであってもよい。ＸＭＬファイルの場合、ファイルには、データ・オブジェクトに関する複数のスクリプト命令又は宣言型定義を含めることができる。例えば、行には、「エンティティ名（ｅｎｔｉｔｙ　ｎａｍｅ）＝Ｌｏｇｉｎ＿Ｅｖｅｎｔ」、「フィールド名（ｆｉｅｌｄ　ｎａｍｅ）＝Ｅｖｅｎｔ＿Ｄａｔｅ」、「フィールドタイプ（ｆｉｅｌｄ　ｔｙｐｅ）＝ＤＡＴＥＴＩＭＥ」などのステートメントが含まれてもよい。この例は、ユーザからのリクエストで、データ・オブジェクト、Ｌｏｇｉｎ＿Ｅｖｅｎｔ、及びＬｏｇｉｎ＿Ｅｖｅｎｔの属性を、日付／時間形式でＥｖｅｎｔ＿Ｄａｔｅという名前で定義する。いくつかの実装では、オブジェクト型をリクエスト内で特定することもできる。オブジェクト型は、システム１００内で参照されるオブジェクトの特定のタイプのインジケータである。オブジェクト型の例は、アカウント、リード、機会、イベントログ、又はチャット・フィードである。いくつかの実装では、データ・オブジェクトの属性は、システム１００の企業又はユーザによって作成又は定義されるカスタム属性であってもよい。いくつかの実装では、リクエストはデータ・オブジェクトに関連する１つ以上のプライマリ・キーを特定する。１つ以上のプライマリ・キーは、データ・オブジェクトの属性又は列である場合がある。１つ以上のプライマリ・キーを指定することにより、指定されたプライマリ・キー列に基づいて、ソートし、クエリすることができる。いくつかの実装では、指定されたプライマリ・キーの少なくとも１つは、データ・オブジェクトの企業ＩＤ属性である。例えば、ｏｒｇ＿ｉｄフィールドは、共有テーブルのプライマリ・キーであり得、テーブルのレコードは、そのｏｒｇ＿ｉｄフィールドに基づいてソートされ得る。従って、ＡＣＭＥに対するレコードは、それらがＥＮＴＥＲＰＲＩＳＥ（企業）のレコードより前に現れるようにソートされる。

【００５０】

ブロック２４０において、システムは、リクエストを処理して、データベース・システム内のデータ・オブジェクトを定義する。いくつかの実装では、企業サーバ１０４又はシステム１００の他の要素は、リクエストを受信し解釈する。いくつかの実装では、リクエストを送信するユーザは、セキュリティについて検査され、リクエストが解釈又はそれに基づいて作動する前に認証される。いくつかの実装では、システムは、ＸＭＬファイルを処理できるアプリケーションなど、リクエストを読み取り、応答又は実行できる１つ以上のアプリケーションを開く。

【００５１】

ブロック２５０において、システムは、データ・オブジェクトを定義するリクエストに基づいて、オブジェクト・スクリプトを生成する。オブジェクト・スクリプトは、リクエストのデータ・オブジェクト及びデータ・オブジェクトの属性に対応する、データベース・システム内の１つ以上のデータベース列を定義する。いくつかの実装では、オブジェクト・スクリプトは、システム１００の１つ以上のアプリケーションによって読み取ることができるデータ・オブジェクト定義、すなわち「ビュー（ｖｉｅｗ）」の形態をとってもよい。いくつかの実装では、オブジェクト・スクリプトはユーザからのリクエストに基づい

て自動的に生成される。例えば、データ・オブジェクトを定義するＸＭＬファイルの形式でユーザ・リクエストを受信すると、システム１００はリクエストを処理し、その後、ＸＭＬデータ・オブジェクト定義を自動的にオブジェクト・スクリプトに変換する。いくつかの実装において、オブジェクト・スクリプトは、データ・オブジェクト及びデータ・オブジェクトの１つ以上の属性を、マルチテナント非リレーショナル・データベースに関連するデータ記述言語におけるデータベース構造として定義する。例えば、非リレーショナル・データベース１１２は、ＳＱＬステートメントが非リレーショナル・データベース１１２上で読み出され、実行されることを可能にするフェニックスのようなアプリケーションと共に機能し得る。従って、アプリケーションは、データ・オブジェクト・リクエストを、非リレーショナル・データベースによって読み取り可能なＳＱＬ用語でオブジェクトを定義する一連のＳＱＬステートメントに変換するように構成されてもよい。例えば、オブジェクト・スクリプトには、「ＷＨＥＮ Ｏｂｊｅｃｔ＿Ｔｙｐｅ ＝ ‘Ｌｏｇｉｎ’, Ｃｏｌｕｍｎ１ ＝ Ｓｏｕｒｃｅ＿ＩＰ ＣＨＡＲ（３２）, Ｃｏｌｕｍｎ２ ＝ Ｅｖｅｎｔ＿Ｄａｔｅ ＤＡＴＥ ＦＲＯＭ Ｂａｓｅ＿Ｐｌａｔｆｏｒｍ＿Ｏｂｊｅｃｔ」などのＳＱＬ又はＳＱＬのようなステートメントを含めることができる。この例では、非リレーショナル・データベース１１２、仮想データベース１１８、又はシステム１００に関連するアプリケーションは、これらのステートメントを理解し、ステートメントに従って非リレーショナル・データベース１１２及び仮想データベース１１８の１つ以上のテーブルに列を追加するように構成されてもよい。いくつかの実装では、オブジェクト・スクリプトは、非リレーショナル・データベース１１２に加えて、又はその代わりに、仮想データベース１１６による処理に対し互換性を有し得る。

【００５２】

ブロック２６０において、システムは、動的仮想テーブルを更新して、オブジェクト・スクリプト内のデータベース列定義に対応する１つ以上の仮想列を含める。仮想データベース１１８の動的仮想テーブルは、データベース層ではなくアプリケーション層で動作するので、物理データベース・スキーマを更新する厳密な要件及び制限を持たない。代わりに、動的仮想テーブルは、１つ以上の列を追加したり、１つ以上の列を削除したり、又はそうでなければ仮想データベース・スキーマを制限なく変更したりできる。

【００５３】

ブロック２７０において、システムは、マルチテナント非リレーショナル・データベース１１２内の共有テーブルの１つ以上の既存の列を更新して、動的仮想テーブルに追加された１つ以上の仮想列と一致させる。いくつかの実装では、１つ以上の列が、新しいデータ・オブジェクトとその属性を表す物理的な非リレーショナル・テーブルに修正、追加、又は削除される。いくつかの実装では、更新される１つ以上の既存の列に関するデータを共有テーブルに書き込むことができる。いくつかの実装において、列を更新することは、プット（Ｐｕｔ）操作、削除（Ｄｅｌｅｔｅ）操作、チェックアンドプット（ＣｈｅｃｋＡｎｄＰｕｔ）操作、チェックアンドデリート（ＣｈｅｃｋａｎｄＤｅｌｅｔｅ）操作、インクリメント（Ｉｎｃｒｅｍｅｎｔ）操作、ゲット（Ｇｅｔ）操作、及びスキャン（Ｓｃａｎ）操作のような、非リレーショナル・データベースで実行される１つ以上の操作を含む。ＨＢａｓｅのような非リレーショナル・データベースは、変わっていく列の複数の構成が異なる行に存在することができるキー値記憶を可能にするので、このようにしてデータベースを更新することは、物理的な非リレーショナル・テーブルのプロパティに違反しない。データベース層では、各列に明示的に定義されたデータ型が存在するのではなく、リレーショナル・データベースに存在するように、その代わりにバイトが存在し、任意の列がアプリケーション層で再定義されることがある。従って、いくつかの実装では、列を下にシフトすることによって列を削除する可能性があり、新しいテーブル全体を作成する必要はない。いくつかの実装では、仮想テーブルの列を非リレーショナル・テーブルにコピーすることにより、新しい列を追加できる。

【００５４】

いくつかの実装では、システムは、マルチテナント非リレーショナル・データベースの共

有テーブルに１つ以上のレコードを追加する。いくつかの実装では、１つ以上のレコードの追加は、１つ以上のイベントで収集されたデータを記憶する企業によって引き起こされるかもしれない。追加されたレコードは、共有テーブルの１つ以上の既存の列又はデータ・オブジェクトに関連付けられる。例えば、レコードのＯｂｊｅｃｔ＿ＩＤフィールドは、どのデータ・オブジェクトがレコードに対応するかを決定する。

【００５５】

図３、図４、図５、及び図６は、ベース・プラットフォーム・オブジェクト方法の前に作成されるデータ・オブジェクトの例を示し、次いで、非リレーショナル・データベースにおいてベース・プラットフォーム・オブジェクト方法を用いて作成されるデータ・オブジェクトの例を示す。

【００５６】

図３は、この出願に記載される方法を使用する前に、非リレーショナル・データベースの物理テーブルに追加されるデータ・オブジェクトの例を示す。「ＬＯＧＩＮ　ＥＶＥＮＴ」という名前のデータ・オブジェクト３１０は、表形式で示される。これには、ＴＥＮＡＮＴ＿ＩＤ、ＣＲＥＡＴＥＤ＿ＤＡＴＥ　などのいくつかの属性が含まれる。各属性は、ＣＨＡＲタイプの１５文字の文字列を表すＣＨＡＲ（１５）のように、その属性のデータ・タイプ又はデータ・フォーマットに対応する。ＨＢａｓｅ内の共有非リレーショナル・テーブル３２０は、データ・オブジェクト３１０を含む。共有テーブル３２０の各列は、データ・オブジェクト３１０の１つの属性に対応する。各行には、データ・オブジェクトの列属性に関連するデータが入力される。例えば、行は、「１２３」のＴＥＮＡＮＴ＿ＩＤ、「１１／１５／２０１４」のＣＲＥＡＴＥＤ＿ＤＡＴＥなどを持ち、ユーザのログインを表す。

【００５７】

図３の列の名前を変更、変更、又は削除する必要がある場合は、その下にあるＨＢａｓｅテーブルを変更する必要がある。共有テーブル３２０は、データベース内のテーブルの物理的なレイアウトを表し、テーブルのダウンタイムを引き起こすことなしに、物理テーブルへの変更を行うことはできない。例えば、テーブルの維持管理者は、ＬＯＧＩＮ＿ＴＹＰＥのデータ・タイプをＣＨＡＲ（１）からＣＨＡＲ（４）に変更することを決めるかもしれない。これを行うには、テーブルを停止する必要があり、更新が実行されるまでログイン・イベントを記録できない。これは、このアプリケーションのＢＰＯ方法なしでテーブルを設計する非動的方法の欠点である。

【００５８】

図４は、いくつかの実装によって実行される非リレーショナル共有テーブルのための動的スキーマの例を示す。ＢＰＯテーブル４１０は、ＨＢａｓｅのマルチテナント、共有、動的、非リレーショナルテーブルである。いくつかの実装において、ＢＰＯテーブル４１０は、テナント又は企業を一意的に特定するためのＴＥＮＡＮＴ＿ＩＤ属性、インスタンスがテーブルに記憶されている特定のデータ・オブジェクトを特定するためのＯＢＪＥＣＴ＿ＴＹＰＥ属性、データ・オブジェクトのインスタンスが作成された時刻を特定するためのＣＲＥＡＴＥＤ＿ＤＡＴＥ属性、及びデータ・オブジェクト・インスタンスの作成者を特定するためのＣＲＥＡＴＥＤ＿ＢＹ属性のうちの１つ以上を含む。いくつかの実装において、ＯＢＪＥＣＴ＿ＴＹＰＥ属性は、参照されるシステム１００内の特定のオブジェクトを指定するコードを特定する。例えば、ログイン・イベント・オブジェクトが「ＬＧＮ」のＯＢＪＥＣＴ＿ＴＹＰＥを有する場合、ついで、レコードが「ＬＧＮ」のＯＢＪＥＣＴ＿ＴＹＰＥを有するＢＰＯテーブル４１０に現れるとき、レコードがログイン・イベントに関連することが確認できる。

【００５９】

図５は、いくつかの実装に従って実行される、動的非リレーショナル共有テーブルに記憶されるいくつかのデータ・オブジェクト定義の例を示す。データ・オブジェクト定義は、非リレーショナル・データベースにデータ・オブジェクト列を構築するための命令を含む、オブジェクト・スクリプト、すなわち「ビュー」でコード化されてもよい。ログイン・

イベント・オブジェクト５１０は、ＢＰＯテーブル４１０の上に定義されたＢＰＯであり、これは、ログイン・イベントのオブジェクト型が指定された場合、図４に定義されたＢＰＯテーブル４１０が、ログイン・イベント・オブジェクト５１０の属性に関連する列と共に拡張されることを意味する。いくつかの実装では、フェニックスなどの１つ以上のアプリケーションは、特定のオブジェクト型に関連するオブジェクトを決定し、次に、どの仮想テーブル及び「ビュー」がベース・テーブル４１０の上に定義され、列を拡張するように構成される。次に、レコードは、非リレーショナル共有テーブルの行に追加され、これは、オブジェクトの属性に関連するすべてのデータを、対応する列のそれぞれについてバイト単位で入力する。フィールド変更イベントオブジェクト５２０は、また、ＢＰＯテーブル４１０の上に定義されたＢＰＯである。

【００６０】

ログイン・イベント・カスタムＢＰＯ５３０は、カスタムＢＰＯの例である。特に、カスタムＢＰＯ５３０は、非リレーショナル・データベースを使用して、特定の顧客又は企業のためにログイン・イベント・オブジェクト５１０の上に定義されたカスタム・ベース・プラットフォーム・オブジェクトである。これは、ベース・テーブル４１０の上に定義されているログイン・イベント５１０オブジェクトに加えて、指定された顧客がレコード内のデータによって特定される場合、列は、ログイン・イベント・カスタムＢＰＯ　５３０属性を含むようにさらに拡張されることを意味する。いくつかの実装において、固有の企業ＩＤは、カスタムＢＰＯオブジェクトの目的のために顧客を特定することがある。この例では、カスタムＢＰＯ　５３０は、顧客のＡｃｍｅ社（Ａｃｍｅ　Ｃｏｒｐ．）に関連する。顧客に固有の１つ以上の属性が定義されることがある。いくつかの実装では、共有テーブルの１つ以上の列に関連するデータを集めるように構成可能な１つ以上の相関属性（Ｃｏｒｒｅｌａｔｉｏｎ　ａｔｔｒｉｂｕｔｅｓ）が定義されることがある。例えば、顧客は、ユーザが実行しているすべてのログインを収集している可能性がある。Ｃｏｒｒｅｌａｔｉｏｎ＿ＩＤ属性は、顧客のすべてのログインを関連付け、カスタム・フィールドとして保存するために使用される。他の顧客はこのフィールドを見ることはなく、この特定の顧客のみが見えることになる。ログインが保存されている場合、顧客は、特定のログイン・イベント又は時間に複数の顧客をマッチさせることができるこの特別な列を見ることができる。ログイン・イベント・カスタムＢＰＯ　５４０は、ログイン・イベント・オブジェクト５１０上で、特に顧客エンタープライズ・インク（Ｅｎｔｅｒｐｒｉｓｅ　Ｉｎｃ．）に対して定義された同様のカスタムＢＰＯである。

【００６１】

図６は、いくつかの実装に従って実行される、非リレーショナル・データベースにおける共有テーブルの物理的レイアウトの例を示す。これは、動的ベース・テーブル４１０及び図５のデータ・オブジェクト定義から生じる物理テーブルであってもよい。まず、ベース・テーブル４１０に関連する列が追加され、データが読み込まれる。最初の行６１０は、このレコードに関連する企業としてＡｃｍｅ　Ｃｏｒｐを指定するＴＥＮＡＮＴ＿ＩＤと、ログイン・イベント・オブジェクト５１０を参照する「ＬＧＮ」を指定するＯＢＪＥＣＴ＿ＴＹＰＥとを有する。また、ベース・テーブル４１０のＣＲＥＡＴＥＤ＿ＤＡＴＥ列及びＣＲＥＡＴＥＤ＿ＢＹ列を有している。フェニックスなどの１つ以上のアプリケーションは、オブジェクト型「ＬＧＮ」を処理し、ログイン・イベント・オブジェクト５１０がこのレコードの列を拡張すべきであると判断する。さらに、１つ以上のアプリケーションは、ＴＥＮＡＮＴ＿ＩＤ「ＡＣＭＥ」を処理し、カスタムＢＰＯ　５３０が、顧客Ａｃｍｅ　Ｃｏｒｐ．に特有の列をさらに拡張すべきであると判断する。アプリケーションは、このレコードのデータをすべて含む仮想テーブルを更新し、これには、ログイン・イベント属性に関連した列と、アクメ・コープ（Ａｃｍｅ　Ｃｏｒｐ．）に関連したカスタム・ログイン・イベント属性に関連した列が含まれる。次に、アプリケーションは、仮想テーブルを使用して、マルチテナント非リレーショナル・データベースの共有テーブルの列を更新し、仮想テーブルに追加された仮想列データと一致させる。従って、行６１０の物理的レイアウトは、バイト形式で１０列すべてを含む仮想テーブルのデータで更新される

。このデータベースの物理スキーマを更新する代わりに、既存の列はバイト単位のデータを含むように変更される。これにより、複数のテナントの属性やデータを変えたオブジェクトを１つの単一の共有テーブルに記憶することができる。いくつかの実装では、上述の記憶プロセスは、リアルタイム又は実質的にリアルタイムで起こる。いくつかの実装では、この物理レイアウトを使用して記憶されたレコードのデータ・オブジェクトを決定し、それを読み出して処理するように構成された１つ以上のアプリケーションに従って、データを照会することができる。

【００６２】

図７は、いくつかの実装に従って実行される、マルチテナント非リレーショナル・データベース・システムのためのカスタム・ベース・プラットフォーム・オブジェクトを生成するための方法７００の一例のフローチャートを示す。最初に、図２からのステップ２１０及び２２０が実行され、各々が複数のレコードを有する複数の企業と関連するマルチテナント非リレーショナル・データベースを保持し、レコードと関連する動的仮想テーブルを保持する。

【００６３】

ブロック７１０において、システムは、企業ＩＤ及びカスタム・データ・オブジェクトの属性を特定して、データベース・システム内のカスタム・データ・オブジェクトを定義するための、企業の１つからのリクエストを受け取る。いくつかの実装では、リクエストはオブジェクト型属性を含んでもよい。リクエストは、カスタム・データ・オブジェクトの少なくとも１つ以上の属性を特定する。いくつかの実装では、ユーザからのリクエストはクライアント・システム１０８から来る。いくつかの実装では、リクエストは企業サーバ１０４から来る。ユーザは、企業又は企業の代表的なメンバー、システム１００の開発者又は保守者、マルチテナント非リレーショナル・データベース１１２の開発者又は保守者、又は他のユーザであってもよい。いくつかの実装において、リクエストは宣言型言語の１つ以上のドキュメントの形式をとる。いくつかの実装においては、企業ＩＤに関連する企業によってリクエスト又はドキュメントが作成される場合がある。いくつかの実装では、リクエストはＸＭＬ又はＪＳＯＮファイルである可能性がある。ＸＭＬファイルの場合、ファイルには、データ・オブジェクトに関する複数のスクリプト命令又は宣言型定義を含めることができる。いくつかの実装において、リクエストは、カスタム・データ・オブジェクトが既存のデータ・オブジェクトの上の拡張（ｅｘｔｅｎｓｉｏｎ）であるべきことを指定するかもしれない。例えば、カスタム・ログイン・イベント・オブジェクトは、標準のログイン・イベント・オブジェクト上で指定及び定義され、特定の企業に興味深い追加の属性を追加できる。いくつかの実装では、リクエストは、カスタム・データ・オブジェクトに関連する１つ以上のプライマリ・キーを特定する。１つ以上のプライマリ・キーは、カスタム・データ・オブジェクトの属性又は列である可能性がある。１つ以上のプライマリ・キーを指定することにより、レコードをソートし、特定されたプライマリ・キーの列に基づいて照会することができる。いくつかの実装では、指定されたプライマリ・キーの１つは、データ・オブジェクトの企業ＩＤ属性である。例えば、ｏｒｇ＿ｉｄフィールドは、共有テーブルのプライマリ・キーの１つであり、テーブルのレコードは、そのｏｒｇ＿ｉｄフィールドに基づいてソートできる。従って、ＡＣＭＥのレコードは、それらがＥＮＴＥＲＰＲＩＳＥのレコードより前に現れるようにソートされる。

【００６４】

ブロック７２０において、システムは、企業からのリクエストを処理して、データベース・システム内のカスタム・データ・オブジェクトを定義する。いくつかの実装では、企業サーバ１０４又はシステム１００の他の要素は、リクエストを受信し解釈する。いくつかの実装では、リクエストを送信する企業、又は企業ＩＤに指定された企業は、セキュリティについて検査され、リクエストを解釈又は処理する前に認証される。いくつかの実装では、システムは、ＸＭＬファイルを処理できるアプリケーションなどの、リクエストを読み取り、応答又は実行できる１つ以上のアプリケーションを開く。

【００６５】

ブロック７３０において、システムは、カスタム・データ・オブジェクトを定義するリクエストに基づいて、企業に関連するカスタム・オブジェクト・スクリプトを生成する。カスタム・オブジェクト・スクリプトは、リクエストのカスタム・データ・オブジェクト及びカスタム・データ・オブジェクトの属性に対応する、データベース・システム内の１つ以上のデータベース列を定義する。いくつかの実装では、カスタム・オブジェクト・スクリプトは、１つ以上の標準データ・オブジェクトの拡張であることを指定する。いくつかの実装では、カスタム・オブジェクト・スクリプトは、システム１００の１つ以上のアプリケーションによって読み取ることができるカスタム・データ・オブジェクト定義又はカスタム「ビュー」の形態をとってもよい。いくつかの実装では、カスタム・オブジェクト・スクリプトは、ユーザからのリクエストに基づいて自動的に生成される。例えば、カスタム・データ・オブジェクトを定義するＸＭＬファイルの形式でユーザ・リクエストを受信すると、システム１００はリクエストを処理し、次に、ＸＭＬカスタム・データ・オブジェクト定義を自動的にカスタム・オブジェクト・スクリプトに変換する。いくつかの実装において、カスタム・オブジェクト・スクリプトは、カスタム・データ・オブジェクト及びカスタム・データ・オブジェクトの１つ以上の属性を、マルチテナント非リレーショナル・データベースに関連するデータ記述言語におけるデータベース構造として定義する。いくつかの実装では、オブジェクト・スクリプトは、非リレーショナル・データベース１１２に加えて、又はその代わりに、仮想データベース１１６による処理に対し互換性を有し得る。

【００６６】

ブロック７４０において、システムは、カスタム・オブジェクト・スクリプト内のデータベース列定義に対応する１つ以上の仮想列を含むように、動的仮想テーブルを更新する。仮想データベース１１８の動的仮想テーブルは、データベース層ではなくアプリケーション層で動作するので、物理データベース・スキーマを更新する厳密な要件及び制限を持たない。代わりに、動的仮想テーブルは、１つ以上の列を追加したり、１つ以上の列を削除したり、又は仮想データベース・スキーマを制限なく変更したりできる。

【００６７】

ブロック７５０において、システムは、マルチテナント非リレーショナル・データベース１１２内の共有テーブルの１つ以上の既存の列を更新して、動的仮想テーブルに追加された１つ以上の仮想列とマッチさせる。いくつかの実装では、１つ以上の列が、新しいカスタム・データ・オブジェクトとその属性を表す、物理的な非リレーショナルテーブルに修正、追加、又は削除される。ＨＢａｓｅのような非リレーショナル・データベースは、様々な列の複数の構成が異なる行に存在することができるキー値記憶を可能にするので、これは物理的な非リレーショナル・テーブルのプロパティに違反しない。データベース層では、各列に明示的に定義されたデータ型が存在するのではなく、リレーショナル・データベースに存在するように、その代わりにバイトが存在し、任意の列がアプリケーション層で再定義されることがある。従って、いくつかの実装では、列を下にシフトすることによって列を削除する可能性があり、新しいテーブル全体を作成する必要はない。いくつかの実装では、仮想テーブルの列を非リレーショナル・テーブルにコピーすることにより、新しい列を追加できる。

【００６８】

ブロック７６０において、システムは、カスタム・データ・オブジェクトに対応する企業ＩＤに関連しない企業のために、共有テーブルの既存の列へのアクセスを制限する。いくつかの実装では、１つ以上のアプリケーションが、１つ以上のカスタム・データ・オブジェクト、又はカスタム・データ・オブジェクトの１つ以上の属性、又はデータ・オブジェクトに関連する共有テーブルの１つ以上の列に関して、１つ以上の企業のパーミッションとアクセス権を変更することがある。いくつかの実装では、アクセスを制限することは、企業ＩＤに関連する企業へのアクセスのみを提供し、他の企業へのアクセスを提供しないことを構成する。いくつかの実装において、アクセスを制限することは、企業ＩＤに関連しないマルチテナント非リレーショナル・データベース内の企業へのアクセスを肯定的に

制限することを構成する。
【００６９】
いくつかの実装では、システムは、共有テーブルの更新された１つ以上の既存の列、又はカスタム・データ・オブジェクトに関連する１つ以上の行又はレコードを、１つ以上のプライバシー設定に関連付ける。いくつかの実装では、プライバシー設定は、１つ以上の既存の列が１つ以上の企業への表示可能性を決定する。いくつかの実装では、より細かいプライバシー設定により、他の企業又は特定のユーザがその企業に関連する１つ以上のデータ・オブジェクト、列、又は行を見ることができるように、企業制御アクセスが与えられることがある。
【００７０】
図８は、いくつかの実装に従って実行される、アクセス制御を介して共有テーブル又はスキーマのサブセットを動的にプロビジョニングするための方法８００の一例のフローチャートを示す。
【００７１】
ブロック８１０において、システムは、各々が複数のレコードを有する、複数の企業と関連するマルチテナント非リレーショナル・データベースを保持する。このステップは、図２のステップ２１０と同一である。
【００７２】
ブロック８２０において、システムは、レコードに関連する共有テーブルであるマルチテナント非リレーショナル・データベース内の共有テーブルを保持する。いくつかの実装では、この共有テーブルは、レコードに関連する標準データ・オブジェクト及びカスタム・データ・オブジェクトを変える１つ以上のレコードを含む。本明細書で説明の技術に従って、異なるレコードに対して異なる数の列が存在してもよい。
【００７３】
ブロック８３０において、システムは、共有テーブルの列に関して、１つ以上の企業に対するパーミッションを特定する共有テーブルに対するアクセス制御を提供する。いくつかの実装では、パーミッション・リストがシステム１００に記憶される。パーミッション・リストは、非リレーショナル・データベースの共有テーブルに関連付けられ、データベースの企業に関するデータ・オブジェクト又は列に対するパーミッションをリストする。いくつかの実装では、システムはパーミッション・リストを検索して、企業に対する現在のパーミッション又はデータ・オブジェクトに対する現在のパーミッションを決定できる。いくつかの実装では、ユーザ・インターフェースは企業サーバ１０４又はクライアント・システム１０８に設けられる。ユーザ・インターフェースは、共有テーブルの１つ以上の列又はデータ・オブジェクトに関して、１つ以上の企業に対する現在のアクセス制御及びパーミッションを表示してもよい。また、ユーザ・インターフェースは、共有テーブルのパーミッションに関連する対話型要素を含んでもよく、これにより、ユーザは、最小限の労力で粒度の細かいアクセス制御を迅速に指定することができる。いくつかの実装において、ユーザ・インターフェースは、粒度の細かいレベル（ｇｒａｎｕｌａｒ　ｌｅｖｅｌ）でユーザへのパーミッションを定義するための管理コンソール（ａｄｍｉｎｉｓｔｒａｔｉｖｅ　ｃｏｎｓｏｌｅ）を提供してもよい。いくつかの実装では、管理コンソールは、特定の企業にある標準又はカスタム・データ・オブジェクトへのアクセスを与えるために、共有テーブルの１つ以上の維持管理者（ｍａｉｎｔａｉｎｅｒｓ）を提供してもよい。いくつかの実装において、アクセス制御を提供することは、１つ以上のデータ・オブジェクトへのアクセスを肯定的に制限することを含み得る。いくつかの実装では、アクセス制御は、共有テーブルの１つ以上の列に対して提供されてもよい。
【００７４】
ブロック８４０において、システムは、共有テーブルの１つ以上の列を企業に提供するリクエストを受信し、このリクエストは、企業に関連する企業ＩＤを特定する。いくつかの実装では、１つ以上の列は、１つ以上の標準データ・オブジェクト又はカスタム・データ・オブジェクトに対応してもよい。いくつかの実装では、リクエストは、共有テーブルの

所有者又は管理者から、又は特別な特権的権利又はモデレーション能力を持つユーザ又は企業から来るかもしれない。いくつかの実装では、システムは、共有テーブルの１つ以上の列に対するパーミッション・メタデータを関連付けるリクエストを追加的に受信する。パーミッション・メタデータは、１つ以上の列へのアクセスをプロビジョニングするための１つ以上のルールを定義するように設定できる。従って、プロビジョン・アクセスへのいくつかのステップは、パーミッション・メタデータを使用して実行可能である。いくつかの実装では、パーミッション・メタデータは、システムの１つ以上のデータベースで定義されてもよい。いくつかの実装では、パーミッション・メタデータは、図９Ｂに示されるように、テナント・データ記憶装置２２内のアプリケーション・メタデータ６６のサブセットとして定義されてもよい。

【００７５】

ブロック８５０において、システムは、共有テーブルの１つ以上の列を企業に提供するリクエストを処理する。いくつかの実装では、企業サーバ１０４又はシステム１００の他の要素は、リクエストを受信し解釈する。いくつかの実装では、リクエストを送信する所有者、維持管理者、又は企業は、リクエストを解釈し又はそれに基づいて行動する前に、セキュリティについて検査され、認証される。いくつかの実装では、システムは、プロビジョニング・アプリケーションなど、リクエストを読み取り、応答し又は実行することができる１つ以上のアプリケーションをオープンする。いくつかの実装では、システムは、アクセス制御を介して１つ以上の列を企業に提供し、それにより、企業に対する可視性（ｖｉｓｉｂｉｌｉｔｙ）、記憶、検索及びその他の能力を可能にする。

【００７６】

ブロック８６０において、システムは、企業ＩＤに関して共有テーブルのプロビジョンされた列に対するパーミッションを変更するために、アクセス制御を更新する。いくつかの実装では、プロビジョニング・アプリケーションは、マルチテナント非リレーショナル・データベースの共有テーブルのパーミッションを変更する。いくつかの実装では、バルク操作（ｂｕｌｋ　ｏｐｅｒａｔｉｏｎ）で複数の企業のパーミッションを同時に修正できる場合がある。いくつかの実装では、バルク操作で複数のデータ・オブジェクトに関連するパーミッションを同時に修正できる。

【００７７】

ブロック８７０において、システムは、共有テーブルの１つ以上の行を更新して、プロビジョンされた列、企業ＩＤに対応する共有テーブルの１つ以上の行を含むようにする。例えば、特定の企業にカスタム・データ・オブジェクトへのアクセスが許可されている場合、次に、共有テーブルの１つ以上の行に、そのカスタム・データ・オブジェクトの属性に関連する列を含めることができる。いくつかの実装では、データ・オブジェクトのオブジェクト・スクリプトはアプリケーションによって処理され、行は図２のブロック２６０及び２７０と同様の方法で更新されてもよく、仮想テーブルは更新され、次いで、非リレーショナル・データベースの共有テーブルの既存の列にバイトが入力される。

【００７８】

開示された技法と併せて、データベース・システム及び企業レベルのソーシャル及びビジネス情報ネットワーク・システムを実装するためのシステム、装置、及び方法を以下に説明する。このような実装は、データベース・システムのより効率的な使用を提供することができる。例えば、データベース・システムのユーザは、データベース内の重要な情報、例えば、プロジェクト又はクライアントに関する情報がいつ変更されたかを容易に知ることができない。このような実装は、そのような変更や他のイベントに関するフィード・トラック更新を提供することができ、それによってユーザに情報を提供し続ける。

【００７９】

例として、ユーザは、例えば、１０００台のコンピュータの可能な販売のような機会のようなＣＲＭレコードの形式でレコードを更新することができる。一旦レコード更新が行われると、レコード更新に関するフィード・トラッキング更新は、次に、自動的に、例えば、フィードで、機会に申し込むもの者（ａｎｙｏｎｅ　ｓｕｂｓｃｒｉｂｉｎｇ　ｔｏ　ｔ

ｈｅ　ｏｐｐｏｒｔｕｎｉｔｙ）又はユーザに提供され得る。従って、更新に関するフィード・トラック更新がフィードを介してマネージャのフィードページ又は他のページに送られるので、ユーザは、機会の変化に関してマネージャに連絡する必要はない。

【００８０】

図９Ａは、いくつかの実装に従って、オンデマンド・データベース・サービスが存在し、使用可能な環境１０の一例のブロック図を示す。環境１０は、ユーザ・システム１２、ネットワーク１４、データベース・システム１６、プロセッサ・システム１７、アプリケーション・プラットフォーム１８、ネットワーク・インターフェース２０、テナント・データ記憶装置２２、システム・データ記憶装置２４、プログラム・コード２６、及びプロセス空間２８を含むことができる。他の実施態様では、環境１０は、これらの構成要素のすべてを有していなくてもよく、及び／又は上述の構成要素の代わりに、又はそれに加えて、他の構成要素を有していてもよい。

【００８１】

ユーザ・システム１２は、データベース・システム１６にアクセスするためにユーザが使用する機械又はシステムのような任意のコンピューティング・デバイス、又は他のデータ処理装置として実装されてもよい。例えば、ユーザ・システム１２のいずれも、携帯電話、スマートフォン、ラップトップ・コンピュータ、又はタブレットのようなハンドヘルド及び／又はポータブル・コンピューティング・デバイスであり得る。ユーザ・システムの他の例としては、ワークステーション及び／又はコンピューティング・デバイスのネットワークのようなコンピューティング・デバイスを含む。図９Ａ（及びより詳細には図９Ｂ）に示すように、ユーザ・システム１２は、ネットワーク１４を介してオンデマンド・データベース・サービスと対話し、オンデマンド・データベース・サービスは、図９Ａの例におけるデータベース・システム１６として実装される。

【００８２】

オンデマンド・データベース・サービスは、例えば、システム１６を用いて実装され、データベース・システムの構築及び／又は保守に必ずしも関与する必要のないユーザに利用可能にされるサービスである。代わりに、データベース・システムは、ユーザがデータベース・システムを必要とするとき、すなわちユーザの要求に応じて、それらの使用のために利用可能であってもよい。いくつかのオンデマンド・データベース・サービスは、マルチテナント・データベース・システム（ＭＴＳ）を形成するために、１つ以上のテナントからの情報を共通のデータベース・イメージのテーブルに記憶することができる。データベース・イメージは、１つ以上のデータベース・オブジェクトを含んでもよい。リレーショナル・データベース管理システム（ＲＤＢＭＳ）又は均等物は、データベース・オブジェクトに対する情報の記憶及び検索を実行することができる。非リレーショナル・データベース管理システム（ＮＲＤＢＭＳ）又は均等物は、データベース・オブジェクトに対する大量の情報の記憶及び高速検索を実行することができる。アプリケーション・プラットフォーム１８は、ハードウェア及び／又はソフトウェア、例えばオペレーティング・システムのような、システム１６のアプリケーションを実行することを可能にするフレームワークであってもよい。いくつかの実装では、アプリケーション・プラットフォーム１８は、オンデマンド・データベース・サービスの提供者、ユーザ・システム１２を介してオンデマンド・データベース・サービスにアクセスするユーザ、又はユーザ・システム１２を介してオンデマンド・データベース・サービスにアクセスするサード・パーティーのアプリケーション開発者によって開発された１つ以上のアプリケーションの作成、管理、及び実行を可能にする。

【００８３】

ユーザ・システム１２のユーザは、それぞれの能力（ｃａｐａｃｉｔｉｅｓ）が異なってもよく、特定のユーザ・システム１２の能力は、現在のユーザに対するパーミッション（パーミッション・レベル）によって完全に決定されてもよい。例えば、セールスマンが特定のユーザ・システム１２を使用してシステム１６と対話している場合、ユーザ・システムは、そのセールスマンに割り当てられた能力を有する。しかし、管理者は、そのユーザ

・システムを使用してシステム１６と対話している間に、そのユーザ・システムは、その管理者に割り当てられた能力を有する。階層的ロール・モデルを持つシステムでは、１つのパーミッション・レベルのユーザは、より低いパーミッション・レベルのユーザによってアクセス可能なアプリケーション、データ、及びデータベース情報にアクセスできるが、より高いパーミッション・レベルのユーザによってアクセス可能な特定のアプリケーション、データベース情報、及びデータにアクセスできないことがある。従って、ユーザのセキュリティ・レベル又はパーミッション・レベル（認可（ａｕｔｈｏｒｉｚａｔｉｏｎ）とも呼ばれる）に依存して、異なるユーザは、アプリケーション及びデータベース情報へのアクセス及び修正に関して異なる能力を有する。

【００８４】

ネットワーク１４は、互いに通信するデバイスのネットワーク又はネットワークの組み合わせである。例えば、ネットワーク１４は、ＬＡＮ（ローカル・エリア・ネットワーク）、ＷＡＮ（ワイド・エリア・ネットワーク）、電話ネットワーク、無線ネットワーク、ポイント・トゥ・ポイント・ネットワーク、スター・ネットワーク、トークン・リング・ネットワーク、ハブ・ネットワーク、又は他の適切な構成のいずれか１つ又は任意の組み合わせであり得る。ネットワーク１４は、ＴＣＰ／ＩＰ（転送制御プロトコル及びインターネットプロトコル）ネットワーク、例えばインターネットと呼ばれることが多いネットワークのグローバルなインターネットワークを含むことができる。インターネットは、本明細書の多くの実施例において使用される。しかしながら、本実装が使用するかもしれないネットワークは、それほど制限されないことを理解されたい。

【００８５】

ユーザ・システム１２は、ＴＣＰ／ＩＰを使用してシステム１６と通信し、より高いネットワークレベルでは、ＨＴＴＰ、ＦＴＰ、ＡＦＳ、ＷＡＰ等のような他の一般的なインターネットプロトコルを使用して通信することができる。ＨＴＴＰが使用される例では、ユーザ・システム１２は、システム１６のＨＴＴＰサーバとの間でＨＴＴＰ信号を送受信する「ブラウザ」と一般に呼ばれるＨＴＴＰクライアントを含むことができる。このようなＨＴＴＰサーバは、システム１６とネットワーク１４との間の唯一のネットワーク・インターフェース２０として実装されてもよいが、他の技法も同様に、又は代わりに使用されてもよい。いくつかの実装では、システム１６とネットワーク１４との間のネットワーク・インターフェース２０は、負荷のバランスを取り、入来ＨＴＴＰリクエストを複数のサーバにわたって均一に分配するためのラウンド・ロビンＨＴＴＰリクエスト・ディストリビュータのような負荷分散機能を含む。少なくともシステム１６にアクセスするユーザに対して、複数のサーバの各々は、ＭＴＳのデータにアクセスするが、他の代わりの代替構成を使用してもよい。

【００８６】

１つの実装において、図９Ａに示されるシステム１６は、ウェブベースのＣＲＭシステムを実装する。例えば、一実施形態では、システム１６は、ＣＲＭソフトウェア・アプリケーションを実装及び実行し、ユーザ・システム１２との間で関連するデータ、コード、フォーム、ウェブ・ページ及びその他の情報を提供し、データベース・システム関連データ、オブジェクト、及びウェブ・ページ・コンテンツを保存及び検索するように構成されたアプリケーション・サーバを含む。マルチテナント・システムでは、複数のテナントのデータは、テナント・データ記憶装置２２内の同じ物理データベース・オブジェクトに記憶され得るが、テナント・データは、典型的には、テナント・データ記憶装置２２の記憶媒体内に配置され、その結果、そのようなデータが明示的に共有されない限り、１つのテナントのデータが他のテナントのデータから論理的に分離され、１つのテナントが他のテナントのデータにアクセスできないように保持される。特定の実施態様では、システム１６は、ＣＲＭアプリケーション以外の、又はそれに加えてアプリケーションを実装する。例えば、システム１６は、ＣＲＭアプリケーションを含む複数のホスト（標準及びカスタム）アプリケーションへのテナント・アクセスを提供することができる。ユーザ（又はサード・パーティーの開発者）アプリケーションは、ＣＲＭを含んでもよいし、含まなくても

よいが、アプリケーション・プラットフォーム１８によってサポートされてもよく、このプラットフォームは、システム１６のプロセス空間における仮想マシンにおけるアプリケーションの作成、１つ以上のデータベース・オブジェクトへのアプリケーションの保存、及びアプリケーションの実行を管理する。

【００８７】

システム１６の要素の一構成は、図９Ａ及び９Ｂに示されるように、ネットワーク・インターフェース２０、アプリケーション・プラットフォーム１８、テナント・データ２３のためのテナント・データ記憶装置２２、システム１６及び場合によってはマルチテナントにアクセス可能なシステム・データ２５のためのシステム・データ記憶装置２４、システム１６の様々な機能を実装するためのプログラム・コード２６、及びアプリケーション・ホスティング・サービスの一部としてアプリケーションを実行するなどのＭＴＳシステム・プロセス及びテナント固有のプロセスを実行するためのプロセス空間２８を含む。システム１６上で実行され得る追加のプロセスは、データベースのインデックス付けプロセスを含む。

【００８８】

図９Ａに示されるシステムのいくつかの要素は、ここで簡単に説明される従来の周知の要素を含む。例えば、各ユーザ・システム１２は、デスクトップ・パーソナル・コンピュータ、ワークステーション、ラップトップ、ＰＤＡ、携帯電話、又は任意の無線アクセス・プロトコル（ＷＡＰ）使用可能デバイス、又はインターネット又は他のネットワーク接続に直接又は間接に接続することができる任意の他のコンピューティング・デバイスを含むことができる。用語「コンピューティング・デバイス（ｃｏｍｐｕｔｉｎｇ　ｄｅｖｉｃｅ）」はまた、本明細書では、単に「コンピュータ（ｃｏｍｐｕｔｅｒ）」とも呼ばれる。ユーザ・システム１２は、典型的には、ＨＴＴＰクライアント、例えば、マイクロソフトのインターネット・エクスプローラ・ブラウザ、ネットスケープのナビゲータ・ブラウザ、オペラのブラウザ、又は携帯電話、ＰＤＡ又は他の無線デバイスなどの場合にはＷＡＰ対応ブラウザを実行し、ユーザ・システム１２のユーザ（例えば、マルチテナント・データベース・システムの加入者）がネットワーク１４を介してシステム１６から利用可能な情報、ページ及びアプリケーションにアクセス、処理、及び閲覧できるようにする。各ユーザ・システム１２はまた、典型的には、コンピューティング・デバイスのディスプレイ（例えば、モニタスクリーン、ＬＣＤディスプレイ、ＯＬＥＤディスプレイなど）上のブラウザによって提供されるＧＵＩと、システム１６又は他のシステム又はサーバによって提供されるページ、フォーム、アプリケーション、及び他の情報と併せて相互作用するための、キーボード、マウス、トラックボール、タッチ・パッド、タッチ・スクリーン、ペンなどのような、１つ以上のユーザ入力装置を含む。従って、本明細書で使用される「ディスプレイ・デバイス（ｄｉｓｐｌａｙ　ｄｅｖｉｃｅ）」は、モニタ又はタッチスクリーン・ディスプレイのようなコンピュータ・システムのディスプレイを指し、デスクトップ・コンピュータ、ラップトップ、タブレット、スマートフォン、テレビのセットトップ・ボックス、又はグーグル・グラス（Ｒ）又は他の人体搭載ディスプレイ装置のようなウェアラブル・デバイスのようなディスプレイ能力を有する任意のコンピュータデバイスを指すことができる。例えば、ディスプレイ装置は、システム１６によってホストされるデータ及びアプリケーションにアクセスし、記憶されたデータに対する検索を実行し、別な方法では、ユーザに提示され得る様々なＧＵＩページとユーザが対話することを可能にするために使用することができる。上述のように、イントラネット、エクストラネット、仮想プライベートネットワーク（ＶＰＮ）、非ＴＣＰ／ＩＰベースのネットワーク、任意のＬＡＮ又はＷＡＮなどの他のネットワークを、インターネットの代わりに、又はインターネットに加えて使用することができるが、実装はインターネットでの使用に適している。

【００８９】

一実施形態によれば、各ユーザ・システム１２及びその構成要素のすべては、インテル・ペンティアム（Ｒ（登録商標））プロセッサなどのような中央処理装置を使用して実行されるコンピュータ・コードを含む、ブラウザなどのアプリケーションを使用してオペレー

タ構成可能である。同様に、システム１６（及び複数が存在する場合のＭＴＳの追加インスタンス）及びその構成要素のすべては、プロセッサ・システム１７を使用して実行するコンピュータ・コードを含むアプリケーションを使用してオペレータ構成可能であってもよく、プロセッサ・システム１７は、インテル・ペンティアム（Ｒ）プロセッサ等を含む中央処理ユニット、及び／又は複数のプロセッサ・ユニットを含むように実装されてもよい。非一時的なコンピュータ読み取り可能媒体は、その上に／中に記憶された命令を有することができ、この命令は、本明細書に説明される任意の実装方法を実行するためにコンピューティング・デバイスによって実行され得るか、又はコンピューティング・デバイスをプログラムするために使用されることができる。Ｗｅｂページ、アプリケーション、及び本明細書で説明される他のデータ及び媒体コンテンツを相互通信し、処理するためにシステム１６を動作させ、構成するための命令を実装するコンピュータ・プログラム・コード２６は、好ましくはダウンロード可能であり、ハードディスクに記憶されるが、プログラム・コード全体又はその一部は、ＲＯＭ又はＲＡＭのような周知の他の揮発性又は不揮発性メモリ媒体又は装置に記憶されてもよく、又はフロッピーディスク、光ディスク、デジタル多用途ディスク（ＤＶＤ）、コンパクト・ディスク（ＣＤ）、マイクロドライブ、磁気光学ディスク、磁気又は光カード、ナノシステム（分子メモリＩＣを含む）、又は命令及び／又はデータを記憶するのに適した任意の他のタイプのコンピュータ読み取り可能媒体又は装置上に提供されてもよい。さらに、プログラム・コード全体、又はその一部は、周知の通信媒体及びプロトコル（例えば、ＴＣＰ／ＩＰ、ＨＴＴＰ、ＨＴＴＰＳ、イーサネット（登録商標）など）を用いて、ソフトウェア・ソースから通信媒体、例えば、インターネット経由、又は他のサーバから送信及びダウンロードされ、又は周知の通信媒体及びプロトコルを用いて周知の他の従来のネットワーク接続（例えば、エクストラネット、ＶＰＮ、ＬＡＮなど）を介して送信されてもよい。また、開示された実装のためのコンピュータ・コードは、クライアント・システム及び／又はサーバ又はサーバ・システム上で、例えば、Ｃ、Ｃ＋＋、ＨＴＭＬ、任意の他のマークアップ言語、Ｊａｖａ（ＴＭ）、ＪａｖａＳｃｒｉｐｔ（登録商標）、ＡｃｔｉｖｅＸ、ＶＢＳｃｒｉｐｔのような任意の他のスクリプト言語、及び周知の多くの他のプログラミング言語で実行することができる任意のプログラミング言語で実現可能であることが理解されよう。（Ｊａｖａ（ＴＭ）は、Ｓｕｎ　Ｍｉｃｒｏｓｙｓｔｅｍｓ，　Ｉｎｃ．の商標である）。

【００９０】

いくつかの実装によれば、各システム１６は、システム１６のテナントとしてのユーザ・システム１２によるアクセスをサポートするために、ユーザ（クライアント）システム１２にウェブ・ページ、フォーム、アプリケーション、データ、及びメディア・コンテンツを提供するように構成される。このように、システム１６は、データが共有されない限り、各テナントのデータを分離しておくためのセキュリティ・メカニズムを提供する。複数のＭＴＳが使用される場合、それらは互いに近接して配置されるか（例えば、単一の建物又はキャンパスに位置するサーバ・ファーム）、又はそれらは互いに離れた場所（例えば、Ａ市に位置する１つ以上のサーバ及びＢ市に位置する１つ以上のサーバ）で分散されるかもしれない。本明細書で使用されるように、各ＭＴＳは、ローカルに、又は１つ以上の地理的位置にわたって分散された１つ以上の論理的及び／又は物理的に接続されたサーバを含むことができる。さらに、用語「サーバ（ｓｅｒｖｅｒ）」は、処理ハードウェア及びプロセス空間を含むシステムのようなあるタイプのコンピューティング・デバイス、メモリ装置又はデータベースのような関連記憶媒体、及び場合によっては、当技術分野で周知のデータベース・アプリケーション（例えば、ＯＯＤＢＭＳ又はＲＤＢＭＳ）を指すことを意味する。また、「サーバ・システム（ｓｅｒｖｅｒ　ｓｙｓｔｅｍ）」及び「サーバ（ｓｅｒｖｅｒ）」は、本明細書ではしばしば互換的に使用されることも理解されたい。同様に、本明細書で説明のデータベース・オブジェクトは、単一のデータベース、分散データベース、分散データベースの集合、冗長なオンライン・バックアップ又はオフライン・バックアップ又は他の冗長性を有するデータベースなどとして実装することができ、分散データベース又は記憶装置ネットワーク及び関連する処理インテリジェンスを含むこ

とができる。
【００９１】
図９Ｂは、図９Ａの要素のいくつかの実装及びこれらの要素間の種々の可能な相互接続の例のブロック図を示す。すなわち、図９Ｂは、環境１０も示している。しかしながら、図９Ｂでは、いくつかの実装におけるシステム１６の要素及び種々の相互接続がさらに示されている。図９Ｂは、ユーザ・システム１２が、プロセッサ・システム１２Ａ、メモリ・システム１２Ｂ、入力システム１２Ｃ、及び出力システム１２Ｄを含み得ることを示す。図９Ｂは、ネットワーク１４及びシステム１６を示す。図９Ｂはまた、システム１６が、テナント・データ記憶装置２２、テナント・データ２３、システム・データ記憶装置２４、システム・データ２５、ユーザ・インターフェース（ＵＩ）３０、アプリケーション・プログラム・インターフェース（ＡＰＩ）３２、ＰＬ／ＳＯＱＬ３４、保存ルーチン３６、アプリケーション設定メカニズム３８、アプリケーション・サーバ５０１～５０Ｎ、システム・プロセス空間５２、テナント・プロセス空間５４、テナント管理プロセス空間６０、テナント記憶空間６２、ユーザ記憶装置６４、及びアプリケーション・メタデータ６６を含むことができることを示す。他の実装では、環境１０は、上述の要素と同じ要素を持たず、及び／又は上述の要素の代わりに、又はそれに加えて、他の要素を有してもよい。
【００９２】
ユーザ・システム１２、ネットワーク１４、システム１６、テナント・データ記憶装置２２、及びシステム・データ記憶装置２４は、図９Ａにおいて上で検討した。ユーザ・システム１２に関して、プロセッサ・システム１２Ａは、１つ以上のプロセッサの任意の組み合わせであってもよい。メモリ・システム１２Ｂは、１つ以上のメモリ・デバイス、短期メモリ、及び／又は長期メモリのいずれかの組み合わせであってもよい。入力システム１２Ｃは、１つ以上のキーボード、マウス、トラックボール、スキャナ、カメラ、及び／又はネットワークへのインターフェースなどの入力装置の任意の組み合わせであってもよい。出力システム１２Ｄは、１つ以上のモニタ、プリンタ、及び／又はネットワークへのインターフェースなどの出力装置の任意の組み合わせであってもよい。図９Ｂに示すように、システム１６は、一組のアプリケーション・サーバ５０として実装されたネットワーク・インターフェース２０（図９Ａの）と、アプリケーション・プラットフォーム１８と、テナント・データ記憶装置２２と、システム・データ記憶装置２４とを含むことができる。また、個々のテナント・プロセス空間５４及びテナント管理プロセス空間６０を含むシステム・プロセス空間５２が示されている。各アプリケーション・サーバ５０は、テナント・データ記憶装置２２及びその中のテナント・データ２３、ならびにシステム・データ記憶装置２４及びその中のシステム・データ２５と通信して、ユーザ・システム１２のリクエストを処理するように構成することができる。テナント・データ２３は、個々のテナント記憶空間６２に分割されてもよく、これは、データの物理的配置及び／又は論理的配置であってもよい。各テナント記憶空間６２内において、ユーザ記憶装置６４及びアプリケーション・メタデータ６６は、各ユーザに対して同様に割り当てられ得る。例えば、ユーザの最新に使用された（ＭＲＵ）項目のコピーは、ユーザ記憶装置６４に記憶されてもよい。同様に、テナントである組織全体に対するＭＲＵ項目のコピーは、テナント記憶空間６２に記憶されてもよい。ＵＩ３０は、ユーザ・インターフェースを提供し、ＡＰＩ３２は、ユーザ・システム１２のユーザ及び／又は開発者に、システム１６に常駐するプロセスにアプリケーションプログラマインターフェースを提供する。テナント・データ及びシステム・データは、１つ以上のＯｒａｃｌｅ（Ｒ）データベースなど、様々なデータベースに保存することができる。
【００９３】
アプリケーション・プラットフォーム１８は、アプリケーション開発者のアプリケーションの作成及び管理をサポートするアプリケーション設定メカニズム３８を含み、これは、例えば、テナント管理プロセス６０によって管理される１つ以上のテナント・プロセス空間５４として、加入者による実行のために、保存ルーチン３６によってテナント・データ記憶装置２２にメタデータとして保存され得る。このようなアプリケーションへの呼び出

しは、ＡＰＩ３２にプログラミング言語スタイルのインターフェース拡張を提供するＰＬ／ＳＯＱＬ３４を使用してコード化することができる。いくつかのＰＬ／ＳＯＱＬ言語実装の詳細な説明は、２０１０年６月１日に発行され、その全体及びすべての目的のために参照により本明細書に組み込まれる、「マルチテナント・オンデマンド・データベース・サービスを介して開発されたアプリケーションへのアクセスを許可するための方法及びシステム（ＭＥＴＨＯＤ ＡＮＤ ＳＹＳＴＥＭ ＦＯＲ ＡＬＬＯＷＩＮＧ ＡＣＣＥＳＳ ＴＯ ＤＥＶＥＬＯＰＥＤ ＡＰＰＬＩＣＡＴＩＯＮＳ ＶＩＡ Ａ ＭＵＬＴＩ－ＴＥＮＡＮＴ ＯＮ－ＤＥＭＡＮＤ ＤＡＴＡＢＡＳＥ ＳＥＲＶＩＣＥ）」という名称の共通の譲渡された米国特許第７，７３０，４７８号で検討されている。アプリケーションへの呼び出しは、１つ以上のシステム・プロセスによって検出することができ、このシステム・プロセスは、呼び出しを行う加入者のためのアプリケーション・メタデータ６６を検索し、そのメタデータを仮想マシンのアプリケーションとして実行することを管理する。

【００９４】

各アプリケーション・サーバ５０は、例えば、異なるネットワーク接続を介してシステム・データ２５及びテナント・データ２３へのアクセスを有するデータベース・システムに通信可能に接続されてもよい。例えば、１つのアプリケーション・サーバ５０１は、ネットワーク１４（例えば、インターネット）を介して接続されてもよく、別のアプリケーション・サーバ５０Ｎ－１は、直接ネットワークリンクを介して接続されてもよく、別のアプリケーション・サーバ５０Ｎは、さらに異なるネットワーク接続によって接続されてもよい。転送制御プロトコル及びインターネットプロトコル（ＴＣＰ／ＩＰ）は、アプリケーション・サーバ５０とデータベース・システムとの間で通信するための典型的なプロトコルである。しかしながら、他のトランスポート・プロトコルが、使用されるネットワーク相互接続に依存してシステムを最適化するために使用され得ることは、当業者に明らかになるだろう。

【００９５】

特定の実装では、各アプリケーション・サーバ５０は、テナントである任意の組織に関連する任意のユーザに対するリクエストを処理するように構成される。何らかの理由で、いつでもサーバ・プールからアプリケーション・サーバを追加及び削除できることが望ましいため、特定のアプリケーション・サーバ５０に対するユーザ及び／又は組織に対するサーバ親和性（ａｆｆｉｎｉｔｙ）は好ましくは存在しない。従って、一実施形態では、負荷分散機能（例えば、Ｆ５ Ｂｉｇ－ＩＰロード・バランサ）を実装するインターフェース・システムが、アプリケーション・サーバ５０とユーザ・システム１２との間で通信可能に接続されて、アプリケーション・サーバ５０にリクエストを分配する。ひとつの実装では、ロード・バランサは、最小接続アルゴリズムを使用して、ユーザ・リクエストをアプリケーション・サーバ５０にルーティングする。ラウンド・ロビンや観測応答時間などの負荷分散アルゴリズムの他の例も使用できる。例えば、ある実装では、同じユーザからの３つの連続したリクエストが３つの異なるアプリケーション・サーバ５０に到達し、異なるユーザからの３つのリクエストが同じアプリケーション・サーバ５０に到達し得る。このようにして、例として、システム１６はマルチテナントであり、システム１６は、異なるユーザ及び組織にわたる異なるオブジェクト、データ及びアプリケーションの保存及びアクセスを処理する。

【００９６】

記憶装置の例として、１つのテナントは、各セールスマンがシステム１６を使って自身の販売プロセスを管理する販売員（ｓａｌｅｓ ｆｏｒｃｅ）を雇用する会社かもしれない。従って、ユーザは、コンタクト・データ、リード・データ、顧客フォローアップ・データ、パフォーマンス・データ、目標、及び進捗データ等を、そのユーザの個人的販売プロセス（例えば、テナント・データ記憶装置２２で）に適用可能である。ＭＴＳ構成の例では、アクセス、閲覧、修正、報告、送信、計算等のすべてのデータ及びアプリケーションは、ネットワーク・アクセス以上のものを有しないユーザ・システムによって保持及びアクセスすることができるので、ユーザは、多くの異なるユーザ・システムのいずれかから

自分の販売努力及びサイクルを管理することができる。例えば、セールスマンが顧客を訪問し、顧客がロビーにインターネット・アクセスを持っている場合、セールスマンは、顧客がロビーに到着するのを待つ間に、その顧客に関する重要な更新を取得することができる。

【００９７】

各ユーザのデータは、各ユーザの雇用主とは無関係に他のユーザのデータから分離されるかもしれないが、いくつかのデータは、テナントである所定の組織に対して、複数のユーザ又は全ユーザによって共有又はアクセス可能な組織全体のデータであるかもしれない。従って、テナント・レベルで割り当てられるシステム１６によって管理されるいくつかのデータ構造があるかもしれないが、他のデータ構造はユーザ・レベルで管理されるかもしれない。ＭＴＳは潜在的な競合者を含む複数のテナントをサポートするかもしれないため、ＭＴＳはデータ、アプリケーション、及びアプリケーションの利用を分離したセキュリティ・プロトコルを持つべきである。また、多くのテナントが独自のシステムを保持するのではなく、ＭＴＳにアクセスすることを選択するかもしれないため、冗長性、使用可能時間（ｕｐ－ｔｉｍｅ）、及びバックアップは、ＭＴＳに実装されるかもしれない付加的な機能である。ユーザ固有のデータ及びテナント固有のデータに加えて、システム１６は、複数のテナント又は他のデータによって使用可能なシステム・レベル・データを保持することもできる。このようなシステム・レベルのデータは、テナント間で共有可能な産業レポート、ニュース、投稿などを含み得る。

【００９８】

特定の実装では、ユーザ・システム１２（クライアント・システムであってもよい）は、アプリケーション・サーバ５０と通信して、テナント・データ記憶装置２２及び／又はシステム・データ記憶装置２４に１つ以上のクエリを送信することを含み得るシステム１６からのシステム・レベル及びテナント・レベルのデータをリクエスト及び更新する。システム１６（例えば、システム１６のアプリケーション・サーバ５０）は、所望の情報にアクセスするように設計された１つ以上のＳＱＬステートメント（例えば、１つ以上のＳＱＬクエリ）を自動的に生成する。システム・データ記憶装置２４は、データベースからリクエストされたデータにアクセスするための照会計画を生成することができる。

【００９９】

各データベースは、一般的に、予め定義されたカテゴリにフィットされたデータを含む論理テーブルの組のようなオブジェクトの集合として見ることができる。「テーブル（ｔａｂｌｅ）」は、データ・オブジェクトの１つの表現であり、本明細書では、いくつかの実装に従ってオブジェクト及びカスタム・オブジェクトの概念的記述を単純化するために使用され得る。「テーブル」及び「オブジェクト（ｏｂｊｅｃｔ）」は、本明細書において互換的に使用され得ることを理解されたい。各テーブルは、通常、表示可能なスキーマ内の列又はフィールドとして論理的に配置された１つ以上のデータ・カテゴリを含む。テーブルの各行又はレコードには、フィールドによって定義される各カテゴリのデータのインスタンスが含まれる。例えば、ＣＲＭデータベースは、名前、住所、電話番号、ファックス番号などの基本的な連絡先情報のためのフィールドを有する顧客を記述するテーブルを含むことができる。別の表は、顧客、製品、販売価格、日付などの情報に関するフィールドを含む、発注書を記載してもよい。

【０１００】

いくつかのマルチテナント・データベース・システムにおいて、すべてのテナントが使用するための標準エンティティ・テーブルが提供される場合がある。ＣＲＭデータベース・アプリケーションの場合、そのような標準エンティティは、各々が事前に定義されたフィールドを含む、ケース、アカウント、コンタクト、リード、及び機会データ・オブジェクトのためのテーブルを含む。用語「エンティティ（ｅｎｔｉｔｙ）」は、本明細書中で「オブジェクト（ｏｂｊｅｃｔ）」及び「テーブル（ｔａｂｌｅ）」と交換可能に使用されてもよいことを理解されたい。いくつかのマルチテナント・データベース・システムでは、テナントはカスタム・オブジェクトの作成及び保存を許可されるか、又は、例えば、カ

スタム・インデックス・フィールドを含む標準オブジェクトのカスタム・フィールドを作成することによって、標準エンティティ又はオブジェクトをカスタマイズすることを許可される場合がある。２０１０年８月１７日に発行され、参照によりその全体及びあらゆる目的で本明細書に組み込まれた、共有に譲渡された、Ｗｅｉｓｓｍａｎ他による米国特許第７，７７９，０３９号「マルチテナント・データベース・システムにおけるカスタム・エントリ及びフィールド（ＣＵＳＴＯＭ　ＥＮＴＲＩＥＳ　ＡＮＤ　ＦＩＥＬＤＳ　ＩＮ　ＡＭＵＬＴＩ－ＴＥＮＡＮＴ　ＤＡＴＡＢＡＳＥ　ＳＹＳＴＥＭ）」は、マルチテナント・データベース・システムにおいて標準オブジェクトをカスタマイズするだけでなく、カスタム・オブジェクトを作成するためのシステム及び方法を教示する。特定の実装では、例えば、すべてのカスタム・エンティティ・データ行は、単一のマルチテナント物理テーブルに記憶され、これは、組織毎に複数の論理テーブルを含むことがある。顧客にとっては、複数の「テーブル（ｔａｂｌｅ）」が実際に１つの大きなテーブルに記憶されていること、あるいは、そのデータが他の顧客のデータと同じテーブルに記憶されていることがトランスペアレントである。

【０１０１】

図１０Ａは、いくつかの実装に従った、オンデマンド・データベース・サービス環境９００のアーキテクチャ構成要素の一例のシステム図を示す。クラウド９０４内に位置するクライアント・マシンは、本明細書で説明されるように、一般に、組み合わされた１つ以上のネットワークを指し、１つ以上のエッジ・ルータ９０８及び９１２を介してオンデマンド・データベース・サービス環境と通信することができる。クライアント・マシンは、上述のユーザ・システム１２の任意の例であり得る。エッジ・ルータは、ファイアウォール９１６を介して１つ以上のコア・スイッチ９２０及び９２４と通信することができる。コア・スイッチは、ロード・バランサ９２８と通信することができ、ロード・バランサ９２８は、ポッド９４０及び９４４のような異なるポッド上にサーバ負荷を分配することができる。ポッド９４０及び９４４は、それぞれ１つ以上のサーバ及び／又は他の計算リソースを含んでもよく、オンデマンドサービスを提供するために使用されるデータ処理及び他の動作を実行してもよい。ポッドとの通信は、ポッド・スイッチ９３２及び９３６を介して行われてもよい。オンデマンド・データベース・サービス環境の構成要素は、データベース・ファイアウォール９４８及びデータベース・スイッチ９５２を介してデータベース記憶装置９５６と通信することができる。

【０１０２】

図１０Ａ及び図１０Ｂに示すように、オンデマンド・データベース・サービス環境にアクセスすることは、種々の異なるハードウェア及び／又はソフトウェア構成要素間で送信される通信を含むことができる。さらに、オンデマンド・データベース・サービス環境９００は、実際のオンデマンド・データベース・サービス環境の単純化された表現である。例えば、図１０Ａ及び１０Ｂには、各タイプの１つ又は２つのデバイスのみが示されているが、オンデマンド・データベース・サービス環境のいくつかの実装は、各タイプの１つ又は多くのデバイスの範囲を含むことができる。また、オンデマンド・データベース・サービス環境は、図１０Ａ及び１０Ｂに示される各デバイスを含む必要はなく、又は図１０Ａ及び１０Ｂに示されない追加デバイスを含むこともできる。

【０１０３】

さらに、オンデマンド・データベース・サービス環境９００内の１つ以上の装置は、同一の物理的装置又は異なるハードウェア上に実装されてもよい。一部の装置は、ハードウェア又はハードウェアとソフトウェアの組み合わせを使用して実装されてもよい。従って、本明細書において使用される「データ処理装置（ｄａｔａ　ｐｒｏｃｅｓｓｉｎｇ　ａｐｐａｒａｔｕｓ）」、「マシン（ｍａｃｈｉｎｅ）」、「サーバ（ｓｅｒｖｅｒ）」及び「デバイス（ｄｅｖｉｃｅ）」のような用語は、単一のハードウェア・デバイスに限定されず、むしろ、説明された機能を提供するように構成された任意のハードウェア及びソフトウェアを含む。

【０１０４】

クラウド９０４は、しばしばインターネットを含むデータ・ネットワーク又はデータ・ネットワークの組み合わせを指すことを意図している。クラウド９０４内に位置するクライアント・マシンは、オンデマンド・データベース・サービス環境と通信して、オンデマンド・データベース・サービス環境によって提供されるサービスにアクセスすることができる。例えば、クライアント・マシンは、情報を検索、記憶、編集、及び／又は処理するために、オンデマンド・データベース・サービス環境にアクセスすることができる。

【０１０５】

いくつかの実装では、エッジ・ルータ９０８及び９１２は、クラウド９０４とオンデマンド・データベース・サービス環境９００の他の構成要素との間でパケットをルーティングする。エッジ・ルータ９０８及び９１２は、ボーダー・ゲートウェイ・プロトコル（ＢＧＰ）を採用してもよい。ＢＧＰは、インターネットのコア・ルーティング・プロトコルである。エッジ・ルータ９０８及び９１２は、インターネット上の自律システム間のネットワーク到達性を指定するＩＰネットワーク又は「プレフィックス（ｐｒｅｆｉｘｅｓ）」の表を保持することができる。

【０１０６】

１つ以上の実装において、ファイアウォール９１６は、オンデマンド・データベース・サービス環境９００の内部構成要素をインターネット・トラフィックから保護することができる。ファイアウォール９１６は、一組のルール及び他の基準に基づいて、オンデマンド・データベース・サービス環境９００の内部構成要素へのアクセスをブロック、許可、又は拒否することができる。ファイアウォール９１６は、パケット・フィルター、アプリケーション・ゲートウェイ、ステートフル・フィルター、プロキシ・サーバー、又は他のタイプのファイアウォールのうちの１つ以上として動作することができる。

【０１０７】

いくつかの実装では、コア・スイッチ９２０及び９２４は、オンデマンド・データベース・サービス環境９００内でパケットを転送する大容量スイッチである。コア・スイッチ９２０及び９２４は、オンデマンド・データベース・サービス環境内の異なる構成要素間でデータを迅速にルーティングするネットワーク・ブリッジとして構成されてもよい。

【０１０８】

いくつかの実装では、２つ以上のコア・スイッチ９２０及び９２４の使用は、冗長性及び／又は遅延の低減を提供し得る。いくつかの実装では、ポッド９４０及び９４４は、オンデマンド・データベース・サービス環境によって提供されるコアデータ処理及びサービス機能を実行することができる。各ポッドは、種々のタイプのハードウェア及び／又はソフトウェアのコンピューティング・リソースを含み得る。ポッド・アーキテクチャの一例は、図１０Ｂを参照してより詳細に検討される。

【０１０９】

いくつかの実施態様では、ポッド９４０と９４４との間の通信は、ポッド・スイッチ９３２と９３６を介して行われてもよい。ポッド・スイッチ９３２及び９３６は、ポッド９４０及び９４４と、クラウド９０４内に位置するクライアント・マシンとの間の通信を、例えば、コア・スイッチ９２０及び９２４を介して容易にすることができる。また、ポッド・スイッチ９３２及び９３６は、ポッド９４０及び９４４とデータベース記憶装置９５６との間の通信を容易にすることができる。

【０１１０】

いくつかの実装では、ロード・バランサ９２８は、ポッド９４０と９４４との間にワークロードを分配することができる。ポッド間のオンデマンド・サービス・リクエストのバランスをとることは、リソースの利用を改善し、スループットを増加させ、応答時間を短縮し、及び／又はオーバーヘッドを低減するのに役立つ。ロード・バランサ９２８は、トラフィックを分析し転送するための多層スイッチを含むことができる。

【０１１１】

いくつかの実装では、データベース記憶装置９５６へのアクセスは、データベース・ファイアウォール９４８によって保護されてもよい。データベース・ファイアウォール９４８

は、プロトコル・スタックのデータベース・アプリケーション層で動作するコンピュータ・アプリケーション・ファイアウォールとして作用してもよい。データベース・ファイアウォール９４８は、構造照会言語（ＳＱＬ）インジェクション、データベース・ルートキット、及び不正な情報開示などのアプリケーション攻撃からデータベース記憶装置９５６を保護することができる。

【０１１２】

いくつかの実装において、データベース・ファイアウォール９４８は、ゲートウェイ・ルータに渡す前に、プロキシ・トラフィックに対して１つ以上の形式の逆プロキシ・サービスを使用するホストを含むことができる。データベース・ファイアウォール９４８は、データベース・トラフィックの内容を検査し、特定のコンテンツ又はデータベース・リクエストをブロックすることができる。データベース・ファイアウォール９４８は、ＴＣＰ／ＩＰスタックの上のＳＱＬアプリケーション・レベルで動作し、データベース又はＳＱＬ管理インターフェースへのアプリケーションの接続を管理し、データベース・ネットワーク又はアプリケーション・インターフェースとの間を往復するパケットをインターセプトし（ｉｎｔｅｒｃｅｐｔｉｎｇ）、実施する（ｅｎｆｏｒｃｉｎｇ）。

【０１１３】

いくつかの実装では、データベース記憶装置９５６との通信は、データベース・スイッチ９５２を介して行われてもよい。マルチテナント・データベース記憶装置９５６は、データベース・クエリを処理するための２つ以上のハードウェア及び／又はソフトウェア構成要素を含むことができる。従って、データベース・スイッチ９５２は、オンデマンド・データベース・サービス環境の他の構成要素（例えば、ポッド９４０及び９４４）によって送信されたデータベース・クエリを、データベース記憶装置９５６内の正しい構成要素に導くことができる。

【０１１４】

いくつかの実装では、データベース記憶装置９５６は、多くの異なる組織によって共有されるオンデマンド・データベース・システムである。オンデマンド・データベース・サービスは、マルチテナント・アプローチ、仮想化アプローチ、又は任意の他のタイプのデータベース・アプローチを採用することができる。オンデマンド・データベース・サービスは、図１０Ａ及び図１０Ｂを参照して、より詳細に検討される。

【０１１５】

図１０Ｂは、いくつかの実装に従った、オンデマンド・データベース・サービス環境のアーキテクチャ構成要素の例をさらに説明するシステム図を示す。ポッド９４４は、オンデマンド・データベース・サービス環境９００のユーザにサービスを提供するために使用することができる。いくつかの実装では、各ポッドは、種々のサーバ及び／又は他のシステムを含み得る。ポッド９４４は、１つ以上のコンテンツ・バッチ・サーバ９６４、コンテンツ検索サーバ９６８、クエリ・サーバ９８２、ファイル・サーバ９８６、アクセス制御システム（ＡＣＳ）サーバ９８０、バッチ・サーバ９８４、及びアプリ・サーバ９８８を含む。また、ポッド９４４は、データベース・インスタンス９９０、クイック・ファイル・システム（ＱＦＳ）９９２、及びインデクサ（ｉｎｄｅｘｅｒｓ）９９４を含む。１つ以上の実装において、ポッド９４４内のサーバ間の通信の一部又は全部がスイッチ９３６を介して送信されてもよい。

【０１１６】

コンテンツ・バッチ・サーバ９６４は、ポッドの内部リクエストを処理することができる。これらのリクエストは、長期にわたる場合もあれば、特定の顧客と結びついていない場合もある。例えば、コンテンツ・バッチ・サーバ９６４は、ログ・マイニング、クリーンアップ・ワーク、及びメンテナンス・タスクに関連するリクエストを処理することができる。

【０１１７】

コンテンツ検索サーバ９６８は、クエリ及びインデクサ機能を提供することができる。例えば、コンテンツ検索サーバ９６８によって提供される機能は、ユーザがオンデマンド・

データベース・サービス環境に記憶されたコンテンツを検索することを可能にし得る。
【０１１８】
ファイル・サーバ９８６は、ファイル記憶装置９９８に記憶された情報に対するリクエストを管理することができる。ファイル記憶装置９９８は、ドキュメント、画像、及びベーシック・ラージ・オブジェクト（ＢＬＯＢｓ）などの情報を記憶することができる。ファイル・サーバ９８６を使用して情報に対するリクエストを管理することにより、データベース上のイメージ・フットプリントを低減することができる。
【０１１９】
クエリ・サーバ９８２は、１つ以上のファイル・システムから情報を検索するために使用されてもよい。例えば、クエリ・システム９８２は、アプリ・サーバ９８８から情報に対するリクエストを受信し、次いで、ポッドの外側に位置するＮＦＳ９９６に情報クエリを送信することができる。
【０１２０】
ポッド９４４は、異なる組織が同じデータベースへのアクセスを共有するマルチテナント環境として構成されたデータベース・インスタンス９９０を共有することができる。さらに、ポッド９４４によって提供されるサービスは、種々のハードウェア及び／又はソフトウェアのリソースを必要とすることがある。いくつかの実装において、ＡＣＳサーバ９８０は、データ、ハードウェア・リソース、又はソフトウェア・リソースへのアクセスを制御することができる。
【０１２１】
いくつかの実装において、バッチ・サーバ９８４は、指定された時間にタスクを実行するために使用されるバッチ・ジョブを処理することができる。従って、バッチ・サーバ９８４は、バッチ・ジョブをトリガするために、Ａｐｐサーバ９８８のような他のサーバに命令を送信してもよい。
【０１２２】
いくつかの実装では、ＱＦＳ　９９２は、カリフォルニア州サンタクララのサン・マイクロシステムズ（Ｓｕｎ　Ｍｉｃｒｏｓｙｓｔｅｍｓ（Ｒ））から入手可能なオープン・ソース・ファイルシステムであってもよい。ＱＦＳは、ポッド９４４内で利用可能な情報を記憶しアクセスするための高速アクセス・ファイル・システムとして機能し得る。ＱＦＳ９９２は、いくつかのボリューム管理機能をサポートし、多くのディスクをファイル・システムにグループ化することを可能にする。ファイル・システムのメタデータは、別々のディスクの組上に保持することができ、これは、長いディスク・シークが許容できないストリーミング・アプリケーションにとって有用であり得る。従って、ＱＦＳシステムは、ネットワーク・ファイルシステム９９６及び／又は他の記憶システムに記憶されたデータを特定、検索、移動、及び／又は更新するために、１つ以上のコンテンツ検索サーバ９６８及び／又はインデクサ９９４と通信してもよい。
【０１２３】
いくつかの実装では、１つ以上のクエリ・サーバ９８２は、ポッド９４４の外部に記憶された情報を検索及び／又は更新するために、ＮＦＳ９９６と通信してもよい。ＮＦＳ９９６は、ポッド９４４内に位置するサーバが、ローカル記憶装置がアクセスされる方法と同様の方法で、ネットワークを介してファイルにアクセスするための情報にアクセスすることを可能にする。
【０１２４】
いくつかの実装では、クエリ・サーバ９２２からのクエリは、ロード・バランサ９２８を介してＮＦＳ９９６に送信されてもよく、ロード・バランサ９２８は、オンデマンド・データベース・サービス環境で利用可能な種々のリソースにわたってリソース・リクエストを分配してもよい。また、ＮＦＳ９９６は、ＱＦＳ９９２と通信して、ＮＦＳ９９６に記憶された情報を更新し及び／又はポッド９４４内に位置するサーバによって使用されるようにＱＦＳ９９２に情報を提供することができる。
【０１２５】

いくつかの実装では、ポッドは、１つ以上のデータベース・インスタンス９９０を含んでもよい。データベース・インスタンス９９０は、ＱＦＳ９９２に情報を送信することができる。情報がＱＦＳに送信されるとき、追加のデータベース呼び出しを使用せずに、ポッド９４４内のサーバによって使用するために情報が利用可能である。

【０１２６】

いくつかの実装では、データベース情報はインデクサ９９４に送信されてもよい。インデクサ９９４は、データベース９９０及び／又はＱＦＳ９９２で利用可能な情報のインデックスを提供することができる。インデックス情報は、ファイル・サーバ９８６及び／又はＱＦＳ９９２に提供されてもよい。

【０１２７】

本明細書に説明又は参照される全部ではなく一部の技法は、ソーシャル・ネットワーキング・システム又はソーシャル・ネットワークとして本明細書に参照されるソーシャルネットワーキング・データベース・システムの一部として又は協働して実行される。ソーシャル・ネットワーキング・システムは、ソーシャル・ネットワーキング・システムの利用者として認識される人々間のコミュニケーションを促進するための一般的な方法になっている。ソーシャル・ネットワーキング・システムの一例は、カリフォルニア州サンフランシスコのＳａｌｅｓｆｏｒｃｅ．ｃｏｍ，　Ｉｎｃ．が提供するチャター（Ｃｈａｔｔｅｒ（Ｒ））である。ｓａｌｅｓｆｏｒｃｅ．ｃｏｍ，　Ｉｎｃ．は、ソーシャル・ネットワーキング・サービス、ＣＲＭサービス、及びその他のデータベース管理サービスのプロバイダーであり、それらのいずれも、いくつかの実施態様において本明細書に開示される技法と組み合わせてアクセスし、使用することができる。これらの種々のサービスは、例えば、マルチテナント・データベース・システムの文脈において、クラウド・コンピューティング環境で提供することができる。従って、開示された技法は、ソフトウェアをローカルにインストールせずに、すなわち、クラウドを介して利用可能なサービスと対話するユーザのコンピューティング・デバイス上で実行することができる。開示された実装は、しばしばＣｈａｔｔｅｒ（Ｒ）を参照して記載されるが、当業者は、開示された技法は、Ｃｈａｔｔｅｒ（Ｒ）にも、セールスフォース・ドット・コム株式会社が提供するその他のサービス及びシステムにも限定されず、Ｆａｃｅｂｏｏｋ（Ｒ）、ＬｉｎｋｅｄＩｎ（Ｒ）、Ｔｗｉｔｔｅｒ（Ｒ）、Ｇｏｏｇｌｅ＋Ｒ、Ｙａｍｍｅｒ（Ｒ）、Ｊｉｖｅ（Ｒ）など、種々の他のデータベース・システム及び／又はソーシャル・ネットワーキング・システムの文脈において実装され得ることを理解すべきである。

【０１２８】

ソーシャル・ネットワーキング・システムの中には、組織を含むさまざまな設定で実装できるものもある。例えば、ソーシャル・ネットワーキング・システムは、企業内、例えば企業内若しくは事業提携内のユーザ、又はそのような組織内のユーザのグループを接続するために実装することができる。例えば、Ｃｈａｔｔｅｒ（Ｒ）は、事業組織の部門の従業員ユーザがデータを共有したり、コミュニケーションしたり、互いに協力したりする際に、しばしば組織の事業を含む様々な社会的目的のために使用することができる。マルチテナント・データベース・システムの例において、組織内の各組織又はグループは、本明細書により詳細に説明されるように、システムの各テナントであり得る。

【０１２９】

いくつかのソーシャルネットワークシステムでは、ユーザは、フィード内の項目又はエントリとして提示される情報更新を含む、１つ以上のソーシャル・ネットワーク・フィードにアクセスすることができる。このようなフィード項目は、単一の情報更新又は個々の情報更新の集合を含むことができる。フィード項目は、文字ベースのデータ、オーディオ・データ、画像データ及び／又はビデオ・データを含む種々のタイプのデータを含むことができる。ソーシャル・ネットワーク・フィードは、本明細書で説明されるようなコンピューティング・デバイスのディスプレイのようなディスプレイ装置上のグラフィカルユーザインターフェース（ＧＵＩ）に表示することができる。情報更新は、種々のソースからの様々なソーシャルネットワークデータを含むことができ、オンデマンド・データベース・

サービス環境に保存することができる。いくつかの実装では、開示された方法、装置、システム、及びコンピュータ読み取り可能な記憶媒体は、マルチテナント・データベース環境において使用するように構成又は設計されてもよい。

【０１３０】

いくつかの実装において、ソーシャル・ネットワーキング・システムは、ユーザが個々のユーザ及びユーザのグループを追跡することに加えて、ケース、アカウント、又は機会のようなＣＲＭレコードの形態のデータ・オブジェクトを追跡することを可能にする。本明細書により詳細に説明されるように、データベースに記憶されたレコードの「フォロー（ｆｏｌｌｏｗｉｎｇ）」により、ユーザは、ユーザがレコードにサブスクライブされたときに、そのレコードの進行を追跡することができる。レコードへの更新は、本明細書でレコードへの変更とも呼ばれ、レコード・フィード又はレコードにサブスクライブしたユーザのニュース・フィードのようなソーシャル・ネットワーク・フィード上で起こり且つ記憶され得る情報更新の一種である。レコード更新の例には、レコードのフィールド変更、レコードのステータスの更新、及びレコード自体の作成が含まれる。一部のレコードは公開でアクセス可能であり、任意のユーザがレコードを追跡できる一方で、他のレコードはプライベートであり、そのレコードをフォローするユーザには適切なセキュリティ・クリアランス／パーミッションが必須である。

【０１３１】

情報更新には、特定のレコードとリンクしている場合とリンクしていない場合がある、様々なタイプの更新を含めることができる。例えば、情報更新は、ユーザによって送信されるソーシャル・メディア・メッセージであってもよく、別な方法では、ユーザのアクションに応答して、又はイベントに応答して生成されてもよい。ソーシャル・メディア・メッセージの例には、投稿、コメント、ユーザの個人的好みの「好み（ｌｉｋｅｓ）」及び「嫌い（ｄｉｓｌｉｋｅｓ）」のような表示、ユーザの状態への更新、アップロードされたファイル、及びユーザが送信したソーシャルネットワークデータ又はインターネット上の種々のドキュメント及び／又はウェブ・ページのような他のネットワークデータへのハイパーリンクが含まれる。投稿は、単語、フレーズ、ステートメント、質問、感情表現、及び／又はシンボルのような、英数字又は他の文字ベースのユーザ入力を含むことができる。コメントは、一般的に、投稿に対する応答、あるいは、単語、フレーズ、ステートメント、答え、質問、反応的な感情的表現及び／又はシンボルなどの、他の情報更新に対する応答を指す。マルチメディア・データは、投稿やコメントに含まれたり、リンクしたり、添付したりできる。例えば、投稿は、ＪＰＥＧ画像又はアニメーション画像と組み合わせたテキスト・ステートメントを含むことができる。特定の投稿やコメントに応じて、好き又は嫌いを提出することができる。アップロードされたファイルの例には、プレゼンテーション、ドキュメント、マルチメディア・ファイルなどが含まれる。

【０１３２】

ユーザは、上述のように、レコードにサブスクライブすることによって、レコードをフォローすることができる。ユーザは、他のタイプのデータ・オブジェクト、他のユーザ、ユーザのグループなどの他のエンティティをフォローすることもできる。このようなエンティティに関するフィード・トラッキング更新は、ユーザのニュース・フィードに受信及び含まれる情報更新の一種である。任意の数のユーザは、特定のエンティティを追跡することができ、従って、ユーザのそれぞれのニュース・フィード上で、そのエンティティに関連する情報更新を見ることができる。いくつかのソーシャル・ネットワークでは、ユーザは互いに接続を確立することによって互いに追従し合うことがあり、時にはお互いに「友達になる（ｆｒｉｅｎｄｉｎｇ）」と呼ばれる。このような接続を確立することによって、あるユーザは、他のユーザによって生成され、他のユーザに関して生成され、又は他の方法として他のユーザに関連する情報を見ることができる。例えば、第１のユーザは、第２のユーザによって第２のユーザの個人的なソーシャル・ネットワーク・ページに投稿された情報を見ることができる。このようなパーソナル・ソーシャル・ネットワーク・ページの１つの実装は、例えば、ユーザのプロファイルを表すウェブ・ページの形式における

、ユーザのプロファイル・ページである。一例において、第１のユーザが第２のユーザをフォローしているとき、第１のユーザのニュース・フィードは、第２のユーザのプロファイル・フィードにサブミットされた第２のユーザから投稿を受け取ることができる。ユーザのプロファイル・フィードは、本明細書で、ユーザのプロファイル・ページに表示されるソーシャル・ネットワーク・フィードの一例である、ユーザの「ウォール（ｗａｌｌ）」とも呼ばれる。

【０１３３】

いくつかの実装において、ソーシャル・ネットワーク・フィードは、ソーシャル・ネットワーキング・システムのユーザのグループに固有であり得る。例えば、ユーザのグループは、ニュース・フィードを公開する（ｐｕｂｌｉｓｈ）ことができる。グループのメンバーは、フィード及びグループのパーミッション構成に従って、このグループのフィードを見て、投稿することができる。グループ・コンテキストにおける情報更新には、グループ・ステータス情報の変更も含めることができる。

【０１３４】

いくつかの実装では、一人以上のユーザから入力された投稿又はコメントのようなデータが、特定のユーザ、グループ、オブジェクト、又はソーシャル・ネットワーキング・システム内の他の構成に対してソーシャル・ネットワーク・フィードに提出されるとき、電子メール通知又は他のタイプのネットワーク通信が、ユーザのプロファイル・フィード、ニュース・フィード、又はレコード・フィードのような１つ以上のフィードの中にフィード項目としてデータを含めることに加えて、ユーザ、グループ、又はオブジェクトをフォローするすべてのユーザに送信されてもよい。いくつかのソーシャル・ネットワーキング・システムでは、そのような通知の発生は、より大きな会話の一部を形成する可能性がある公開された入力の最初のインスタンスに限定される。例えば、通知は、最初の投稿に対しては送信されるが、投稿に対するコメントに対しては送信されない。他のいくつかの実装では、そのような情報更新毎に別々の通知が送信される。

【０１３５】

用語「マルチテナント・データベース・システム（ｍｕｌｔｉ－ｔｅｎａｎｔ　ｄａｔａｂａｓｅ　ｓｙｓｔｅｍ）」は、一般に、データベース・システムのハードウェア及び／又はソフトウェアの種々の要素が１つ以上の顧客によって共有され得るシステムを指す。例えば、所定のアプリケーション・サーバは、多数の顧客に対するリクエストを同時に処理することができ、所定のデータベース・テーブルは、より多くの潜在的な顧客のために、フィード項目のようなデータの行を保存することができる。

【０１３６】

「ユーザー・プロファイル（ｕｓｅｒ　ｐｒｏｆｉｌｅ）」又は「ユーザー・プロファイル（ｕｓｅｒ’ｓ　ｐｒｏｆｉｌｅ）」の例は、ソーシャル・ネットワーキング・システム及び／又はデータベース・システムの所定のユーザに関するデータを保存及び保持するように構成されたデータベース・オブジェクト又はオブジェクトのセットである。データは、名前、タイトル、電話番号、写真、経歴サマリー、及びステータス（例えば、ユーザが現在行っていることを記述するテキスト）などの一般的な情報を含み得る。本明細書で述べたように、データは、他のユーザによって作成されたソーシャル・メディア・メッセージを含むことができる。複数のテナントが存在する場合、ユーザは典型的には特定のテナントに関連付けられる。例えば、ユーザは、データベース・サービスを提供するデータベース・システムのテナントである企業のセールスマンであり得る。

【０１３７】

用語「レコード（ｒｅｃｏｒｄ）」は、一般に、値を持つフィールドを有し、データベース・システムに記憶されるデータ・エンティティを指す。レコードの例は、例えば、特定の（実際の又は潜在的な）ビジネス関係又はプロジェクトに関するＣＲＭレコードの形式で、データベース・サービスのユーザによって作成されたデータ・オブジェクトのインスタンスである。レコードは、データベース・サービスによって定義されるデータ構造（標準オブジェクト）又はユーザによって定義されるデータ構造（カスタム・オブジェクト）

を持つことができる。例えば、レコードは、ユーザのビジネス・パートナー又は潜在的なビジネス・パートナー（例えば、クライアント、ベンダー、ディストリビュータなど）に対するものであってもよく、会社全体、子会社、又は会社における連絡先を記述する情報を含むことができる。別の例として、レコードは、既存のパートナーとの機会（例えば、販売の可能性）のようなユーザが取り組んでいるプロジェクト、又はユーザが得ようとしているプロジェクトであり得る。マルチテナント・データベース・システムの１つの実装において、テナントの各レコードは、共有テーブルに記憶された固有の識別子を有する。レコードは、オブジェクトの構造によって定義されるデータフィールド（例えば、特定のデータ・タイプと目的のフィールド）を持つ。レコードには、ユーザが定義したカスタム・フィールドもある。フィールドは、別のレコードであるか、又はそれへのリンクを含むことができ、それによってレコード間の親子関係を提供する。

【０１３８】

用語「ソーシャル・ネットワーク・フィード（ｓｏｃｉａｌ　ｎｅｔｗｏｒｋ　ｆｅｅｄ）」及び「フィード（ｆｅｅｄ）」は、本明細書中で互換的に使用され、一般に、フィード項目又は種々のタイプの情報及びデータを有するエントリの組み合わせ（例えば、リスト）を指す。そのようなフィード項目は、表示されるフィードの一部として提示される関連情報を検索するためにアクセスすることができる１つ以上のデータベース・テーブル、例えば、テーブル内の行として、保存及び保持することができる。用語「フィード項目（ｆｅｅｄ　ｉｔｅｍ）」（又はフィード要素（ｆｅｅｄ　ｅｌｅｍｅｎｔ））は、一般に、ユーザによって提出される投稿のようなフィードで提示され得る情報の項目を指す。ユーザに関する情報のフィード項目は、データベースのユーザのプロファイル・フィードに表示することができ、一方、レコードに関する情報のフィード項目は、例として、データベースのレコード・フィードに表示することができる。プロファイル・フィードとレコード・フィードは、異なるタイプのソーシャル・ネットワーク・フィードの例である。第１のユーザ及びレコードをフォローする第２のユーザは、第１のユーザに関連するフィード項目及び第２のユーザのニュース・フィードに表示するためのレコードを受信することができ、これは別のタイプのソーシャル・ネットワーク・フィードである。いくつかの実装では、追跡された任意の数のユーザ及びレコードからのフィード項目を、特定のユーザの単一のソーシャル・ネットワーク・フィードに組み合わせることができる。

【０１３９】

例えば、フィード項目は、テキスト・データのユーザ生成投稿のようなソーシャル・メディア・メッセージ、及びレコード又はプロファイルへのフィード・トラック更新、例えばレコードのフィールドへの変更であってもよい。フィード・トラッキング更新は、本明細書においてより詳細に説明される。フィードは、ソーシャル・メディア・メッセージとフィード・トラッキング更新の組み合わせであり得る。ソーシャル・メディア・メッセージは、ユーザによって作成されたテキストを含み、他のデータも含むことができる。ソーシャル・メディアのメッセージの例には、投稿、ユーザ・ステータス更新、及びコメントが含まれる。ソーシャル・メディア・メッセージは、ユーザのプロファイル又はレコード用に作成できる。投稿は、いくつかの制限を適用することができるが、様々なユーザ、潜在的に任意のユーザによって作成することができる。一例として、投稿は、ユーザのプロファイル・ページのウォール部分（多数の最近の投稿を含むことができる）又は複数の投稿を含むレコードの部分に作成することができる。投稿は、例えば、ユーザのプロファイル・フィードの一部として、ユーザのプロファイル・ページにＧＵＩで表示されるとき、時系列に編成することができる。投稿とは対照的に、ユーザ・ステータス更新はユーザのステータスを変更し、そのユーザ又は管理者によって行うことができる。レコードは、ステータスを持つこともでき、その更新は、レコードの所有者、又はレコードに対する適切な書き込みアクセス・パーミッションを持つ他のユーザによって提供される。所有者は、単一のユーザ、複数のユーザ、又はグループになる。

【０１４０】

いくつかの実装では、任意のフィード項目に対してコメントを行うことができる。いくつ

かの実装では、コメントは、特定のフィード・トラック更新、投稿、又はステータス更新に明示的に結び付けられたリストとして編成される。いくつかの実施態様では、コメントは、フィード項目の第１の層（階層的意味で）にはリストされないが、特定の第１の層フィード項目から分岐する第２の層としてリストされてもよい。

【０１４１】

「フィード・トラック更新（ｆｅｅｄ　ｔｒａｃｋｅｄ　ｕｐｄａｔｅ）」は、本明細書中では「フィード更新（ｆｅｅｄ　ｕｐｄａｔｅ）」とも呼ばれ、情報更新の１つのタイプであり、一般に、イベントを表すデータを指す。フィード・トラッキング更新は、イベントに応答してデータベース・システムによって生成されたテキストを含み、１つ以上のフィードに含める可能性のある１つ以上のフィード項目として提供される。１つの実装において、データは最初に記憶され、次に、データベース・システムは、後に、そのデータを使用して、イベントを説明するためのテキストを作成することができる。データ及び／又はテキストの両方は、本明細書で使用されるように、フィード・トラック更新であり得る。種々の実装において、イベントはレコードの更新であってもよく、及び／又はユーザによる特定のアクションによってトリガされてもよい。イベントのトリガとなるアクションは、構成可能である。どのイベントが生成されたフィード・トラッキング更新を持ち、どのフィード更新が送信され、ユーザも構成可能である。ソーシャル・メディア・メッセージやその他の種類のフィード更新は、レコードのフィールド又は子オブジェクトとして保存できる。例えば、フィードはレコードの子オブジェクトとして保存できる。

【０１４２】

「グループ（ｇｒｏｕｐ）」は、一般に、ユーザの集合である。いくつかの実装において、グループは、同一又は類似の属性を持つユーザ、又はメンバーシップによって定義されてもよい。いくつかの実装では、本明細書において「グループ・ニュース・フィード（ｇｒｏｕｐ　ｎｅｗｓ　ｆｅｅｄ）」とも呼ばれる「グループ・フィード（ｇｒｏｕｐ　ｆｅｅｄ）」は、グループ内の任意のユーザに関する１つ以上のフィード項目を含む。いくつかの実装では、グループ・フィードは、グループ全体、グループの目的、グループの説明、及びグループのレコード並びにグループに関連して保存された他のオブジェクトに関する情報更新及び他のフィード項目も含む。グループ・レコード更新や、投稿、コメント、好みなどのソーシャル・メディア・メッセージを含む情報更新のスレッドは、グループ会話を定義し、時間の経過とともに変化させることができる。

【０１４３】

「エンティティ・フィード」又は「レコード・フィード」は、一般に、データベース内の特定のレコードに関するフィード項目のフィードを指す。このようなフィード項目には、レコードの変更に関するフィード・トラック更新、及びレコードに関するユーザによる投稿が含まれる。エンティティ・フィードは、任意の種類のフィード項目で構成できる。このようなフィードは、レコードに関連するウェブ・ページのようなページ、例えばレコードのホームページに表示することができる。本明細書で使用される「プロファイル・フィード」又は「ユーザ・プロファイル・フィード」は、一般に、特定のユーザに関するフィード項目のフィードを指す。一例において、プロファイル・フィードのためのフィード項目は、他のユーザが特定のユーザに対して作成又は送信する投稿及びコメント、及び特定のユーザによって行われる状態更新を含む。このようなプロファイル・フィードは、特定のユーザに関連するページに表示することができる。別の例では、プロファイル・フィード内のフィード項目は、特定のユーザによって作られた投稿と、特定のユーザの動作に基づいて開始されたフィード・トラック更新とを含むことができる。

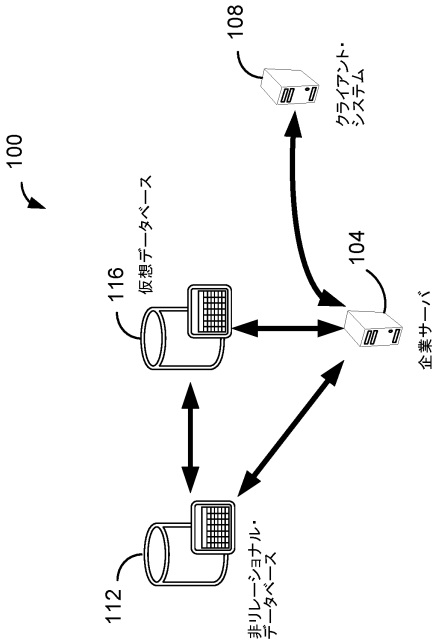
【０１４４】

開示された実装のいくつかは、複数のテナントをサポートすることができるオンデマンド・データベース・サービスのフロント・エンドを提供するアプリケーション・サーバを有するシステムに関して記述されてもよいが、開示された実装は、マルチテナント・データベース又はアプリケーション・サーバへの展開に限定されない。いくつかの実装は、請求された実装の範囲から逸脱することなく、ＯＲＡＣＬＥ（Ｒ）、ＩＢＭによるＤＢ２（Ｒ

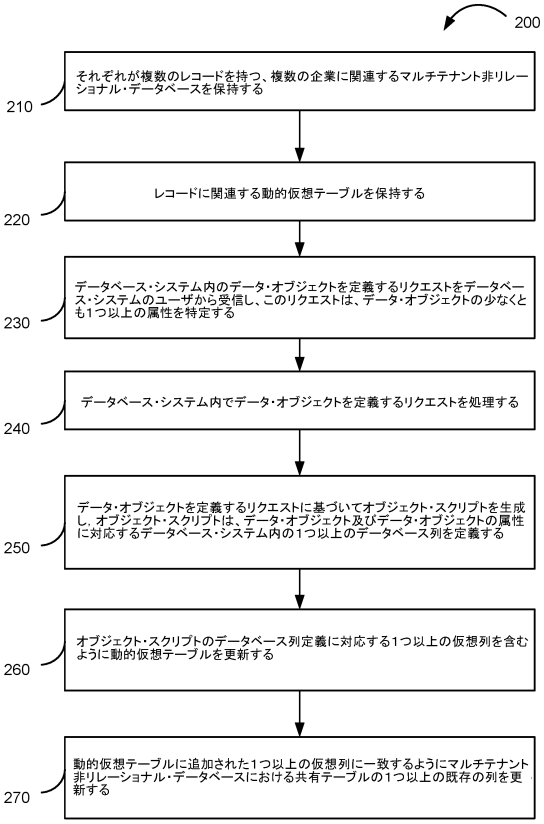
）等のような種々のデータベース・アーキテクチャを使用して実践することができる。
【０１４５】
開示された実装のいくつかは、モジュラー方式又は統合方式でハードウェア及び／又はコンピュータ・ソフトウェアを使用する制御論理の形式で実現可能であることを理解されたい。ハードウェア及びハードウェアとソフトウェアの組み合わせを使用して、他のやり方及び／又は方法が可能である。
【０１４６】
開示された任意の実施形態は、種々のタイプのハードウェア、ソフトウェア、ファームウェア、及びそれらの組み合わせで具体化され得る。例えば、本明細書に開示されるいくつかの技術は、少なくとも部分的に、本明細書で説明される種々のサービス及び動作を実行するためのプログラム命令、状態情報などを含むコンピュータ読み取り可能媒体によって実装されてもよい。プログラム命令の例は、コンパイラによって生成されるようなマシンコードと、サーバ又はインタプリタを使用する他のデータ処理装置のようなコンピューティング・デバイスによって実行され得る高レベルコードを含むファイルの両方を含む。コンピュータ読み取り可能媒体の例には、ハードディスク、フロッピーディスク、及び磁気テープのような磁気媒体；フラッシュメモリ、コンパクト・ディスク（ＣＤ）、又はデジタル多用途ディスク（ＤＶＤ）のような光媒体、磁気光学媒体、及びリード・オンリー・メモリ（「ＲＯＭ」）デバイス及びランダム・アクセス・メモリ（「ＲＡＭ」）デバイスのようなプログラム命令を記憶するように特別に構成されたハードウェア・デバイスが含まれるが、これらに限定されない。コンピュータ読み取り可能媒体は、そのような記憶装置の任意の組み合わせであってもよい。
【０１４７】
本出願に記載される動作及び技法のいずれも、例えばオブジェクト指向技法を用いて、例えばＪａｖａ、Ｃ＋＋又はＰｅｒｌのような任意の適切なコンピュータ言語を用いてプロセッサによって実行されるソフトウェアコードとして実装することができる。ソフトウェアコードは、コンピュータ読み取り可能媒体上に一連の命令又はコマンドとして記憶されてもよい。ソフトウェア／プログラム・コードで符号化されたコンピュータ読み取り可能媒体は、互換装置にパッケージされるか、又は他の装置とは別個に（例えば、インターネットダウンロードを介して）提供されてもよい。このような任意のコンピュータ読取り可能媒体は、単一のコンピューティング・デバイス又はコンピュータ・システム全体上に又はその中に存在してもよく、システム又はネットワーク内の他のコンピュータ読み取り可能媒体の中にあってもよい。コンピュータ・システム又はコンピューティング・デバイスは、本明細書で説明した結果のいずれかをユーザに提供するためのモニタ、プリンタ、又は他の適切なディスプレイを含むことができる。
【０１４８】
本明細書においては、様々な実施例が記載されているが、それらは例示としてのみ提示されており、限定されるものではないことを理解されたい。従って、本出願の広がり及び範囲は、本明細書で説明する実装のいずれによっても制限されるべきではなく、以下の及び後に提出される請求項及びその等価物に従ってのみ規定されるべきである。

【図面】

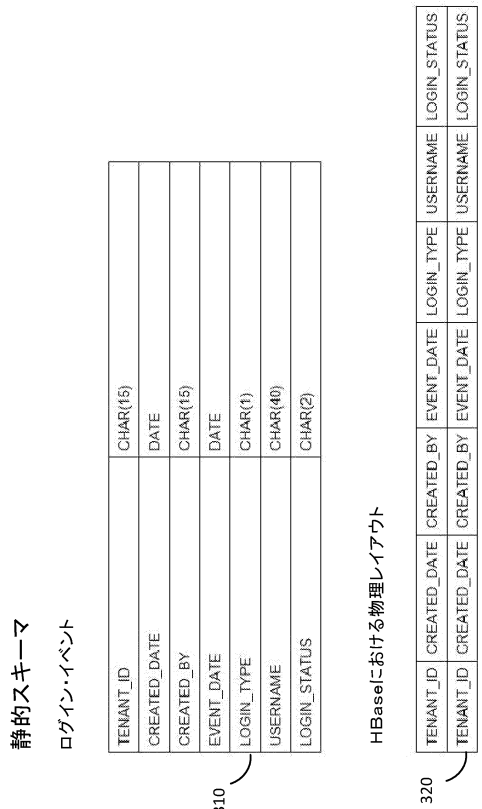
【図１】


100
108
クライアント・システム
仮想データベース
116
104
企業サーバ
112
非リレーショナル・データベース

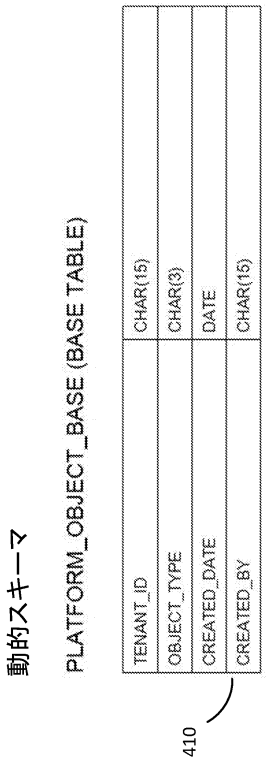
【図２】


200
210
それぞれが複数のレコードを持つ、複数の企業に関連するマルチテナント非リレーショナル・データベースを保持する
220
レコードに関連する動的仮想テーブルを保持する
230
データベース・システム内のデータ・オブジェクトを定義するリクエストをデータベース・システムのユーザから受信し、このリクエストは、データ・オブジェクトの少なくとも1つ以上の属性を特定する
240
データベース・システム内でデータ・オブジェクトを定義するリクエストを処理する
250
データ・オブジェクトを定義するリクエストに基づいてオブジェクト・スクリプトを生成し、オブジェクト・スクリプトは、データ・オブジェクト及びデータ・オブジェクトの属性に対応するデータベース・システム内の1つ以上のデータベース列を定義する
260
オブジェクト・スクリプトのデータベース列定義に対応する1つ以上の仮想列を含むように動的仮想テーブルを更新する
270
動的仮想テーブルに追加された1つ以上の仮想列に一致するようにマルチテナント非リレーショナル・データベースにおける共有テーブルの1つ以上の既存の列を更新する

【図３】


静的スキーマ
ログイン・イベント
TENANT_ID CHAR(15)
CREATED_DATE DATE
CREATED_BY CHAR(15)
EVENT_DATE DATE
LOGIN_TYPE CHAR(1)
USERNAME CHAR(40)
LOGIN_STATUS CHAR(2)
310
HBaseにおける物理レイアウト
TENANT_ID CREATED_DATE CREATED_BY EVENT_DATE LOGIN_TYPE USERNAME LOGIN_STATUS
TENANT_ID CREATED_DATE CREATED_BY EVENT_DATE LOGIN_TYPE USERNAME LOGIN_STATUS
320

【図４】


動的スキーマ
PLATFORM_OBJECT_BASE (BASE TABLE)
TENANT_ID CHAR(15)
OBJECT_TYPE CHAR(3)
CREATED_DATE DATE
CREATED_BY CHAR(15)
410

【図５】

LOGIN_EVENT View

PLATFORM__OBJECT__BASE上に定義される

510

EVENT_DATE	DATE
LOGIN_TYPE	CHAR(1)
USERNAME	CHAR(40)
LOGIN_STATUS	CHAR(2)

FIELD_CHANGE_EVENT View

520

EVENT_DATE	DATE
OLD_VALUE	VARCHAR
NEW_VALUE	VARCHAR
PARENT_OBJECT_ID	CHAR(15)

LOGIN_EVENT view for Customer Acme Corp

特定の顧客のLOGIN__EVENTビュー上に定義される

530

ACME_SYSTEM_ID	CHAR(2)
CORRELATION_ID	CHAR(200)

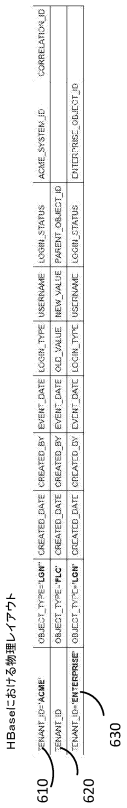
LOGIN_EVENT view for Customer Enterprise Inc

特定の顧客のLOGIN__EVENTビュー上に定義される

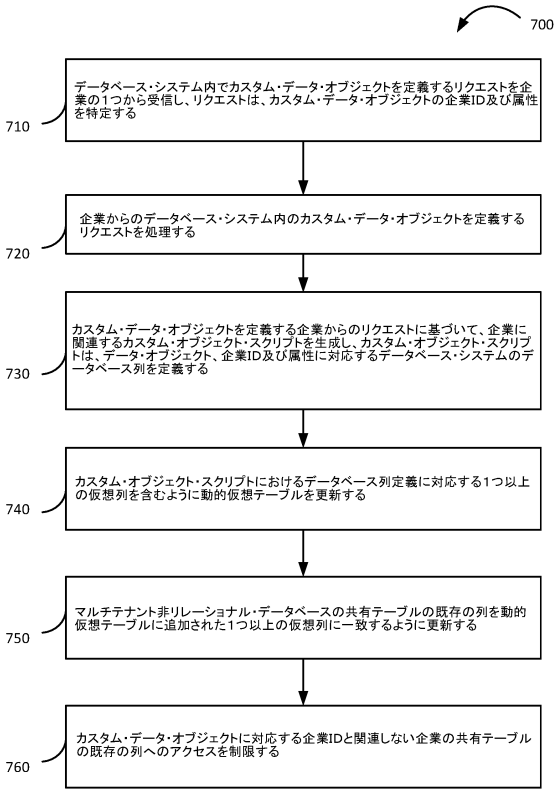
540

ENTERPRISE_OBJECT_ID	CHAR(20)

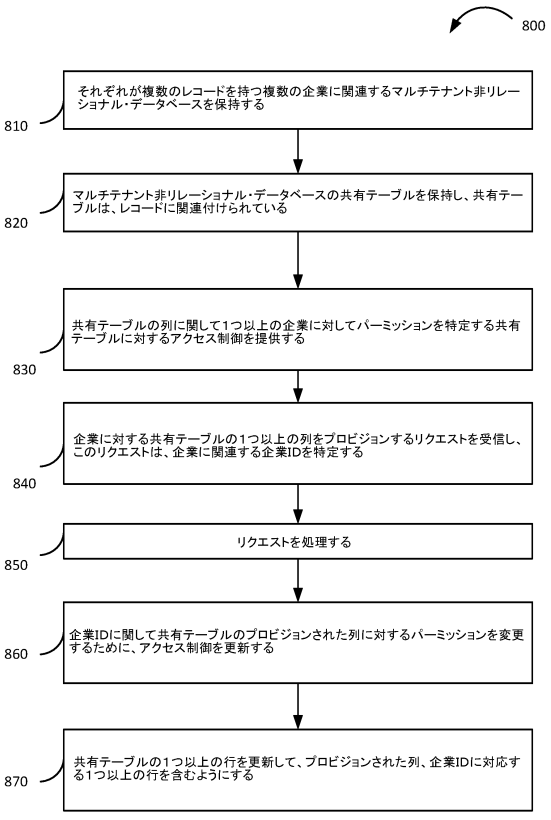
【図６】



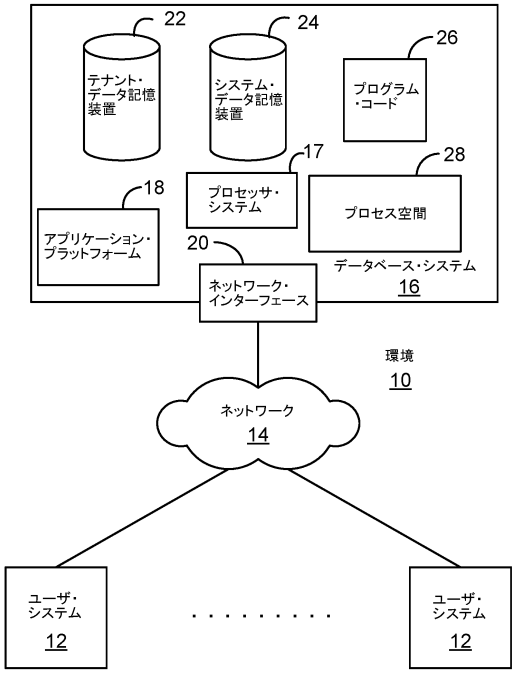
【図７】



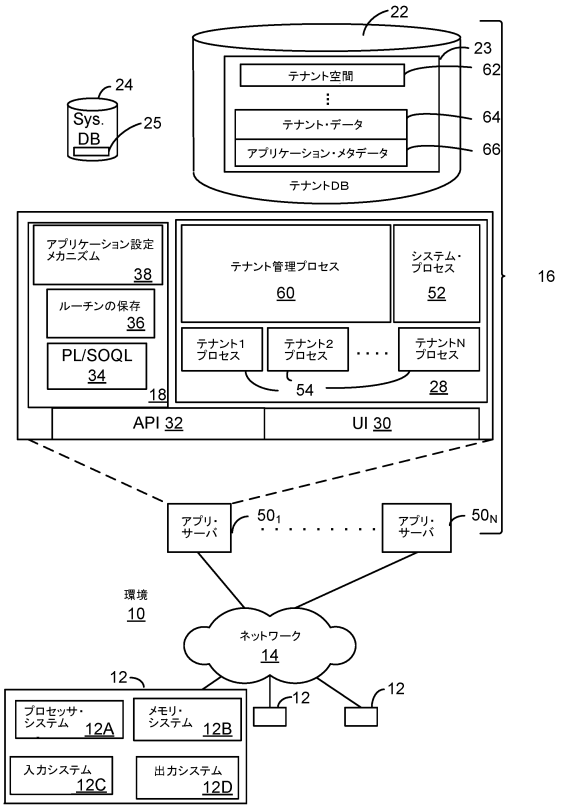
【図８】



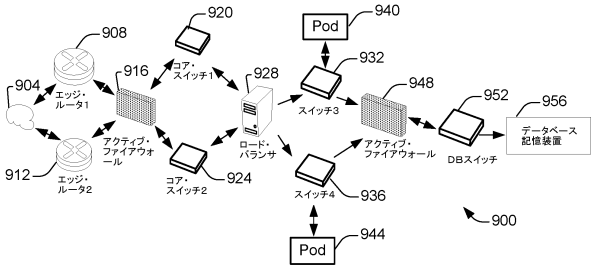
【図９Ａ】



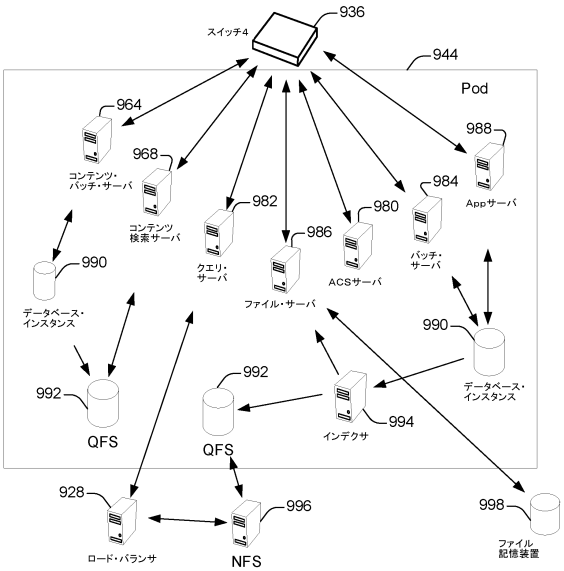
【図９Ｂ】



【図１０Ａ】



【図１０Ｂ】



フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号 15/283,145
(32)優先日 平成28年9月30日(2016.9.30)
(33)優先権主張国・地域又は機関
米国(US)
ォックス サークル ２５８２
(72)発明者 トーマン，アダム
アメリカ合衆国 カリフォルニア州 ９４５９５，ウォールナット クリーク，ウィロー アヴェニュー ４１
(72)発明者 レヴィン，イーライ
アメリカ合衆国 カリフォルニア州 ９４１２１，サン フランシスコ，クレメント ストリート ２４４３，＃５
(72)発明者 フェルナンド，ジャン アシタ
アメリカ合衆国 カリフォルニア州 ９４１１６，サン フランシスコ，アロア ストリート ３０４４
(72)発明者 ジェイン，サマールパン
アメリカ合衆国 カリフォルニア州 ９４５３６，フリーモント，フォークナー コート ４５６５
審査官 後藤 彰
(56)参考文献 米国特許出願公開第２０１５／０１４２７８３（ＵＳ，Ａ１）
米国特許出願公開第２００５／０２２３０２２（ＵＳ，Ａ１）
米国特許出願公開第２０１４／０３８００５１（ＵＳ，Ａ１）
米国特許出願公開第２０１２／０１９１７３５（ＵＳ，Ａ１）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １６／００－１６／９５８