US 20110246524A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2011/0246524 A1**

(43) **Pub. Date: Oct. 6, 2011**

Wu et al.

(54) **SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR PORTAL USER DATA ACCESS IN A MULTI-TENANT ON-DEMAND DATABASE SYSTEM**

(75) Inventors: **Yongsheng Wu**, Redwood City, CA (US); **Alfred Vieira**, Oakland, CA (US); **Punit Jain**, Cupertino, CA (US)

(73) Assignee: **salesforce.com, inc.**, San Francisco, CA (US)

(21) Appl. No.: **13/078,858**

(22) Filed: **Apr. 1, 2011**

Related U.S. Application Data

(60) Provisional application No. 61/320,152, filed on Apr. 1, 2010, provisional application No. 61/320,188, filed on Apr. 1, 2010.

Publication Classification

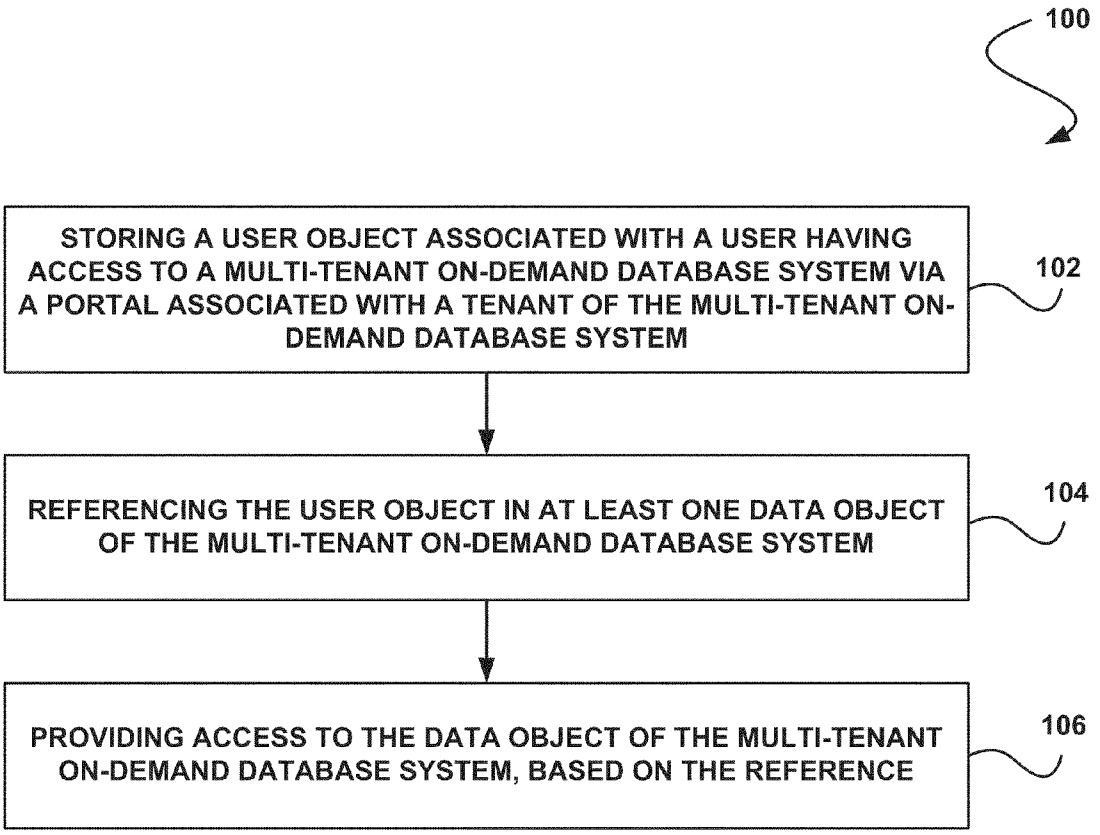
(51) **Int. Cl.**

G06F 17/30	(2006.01)
G06F 7/00	(2006.01)

(52) **U.S. Cl.** **707/781**; 707/E17.001

(57) **ABSTRACT**

In accordance with embodiments, there are provided mechanisms and methods for portal user data access in a multi-tenant on-demand database system. These mechanisms and methods for portal user data access in a multi-tenant on-demand database system can enable embodiments to provide portal-specific user accounts to the multi-tenant on-demand database system which have reduced configuration requirements than users directly accessing the multi-tenant on-demand database system. The ability of embodiments to provide portal-specific user accounts can reduce processing requirements of the database system.



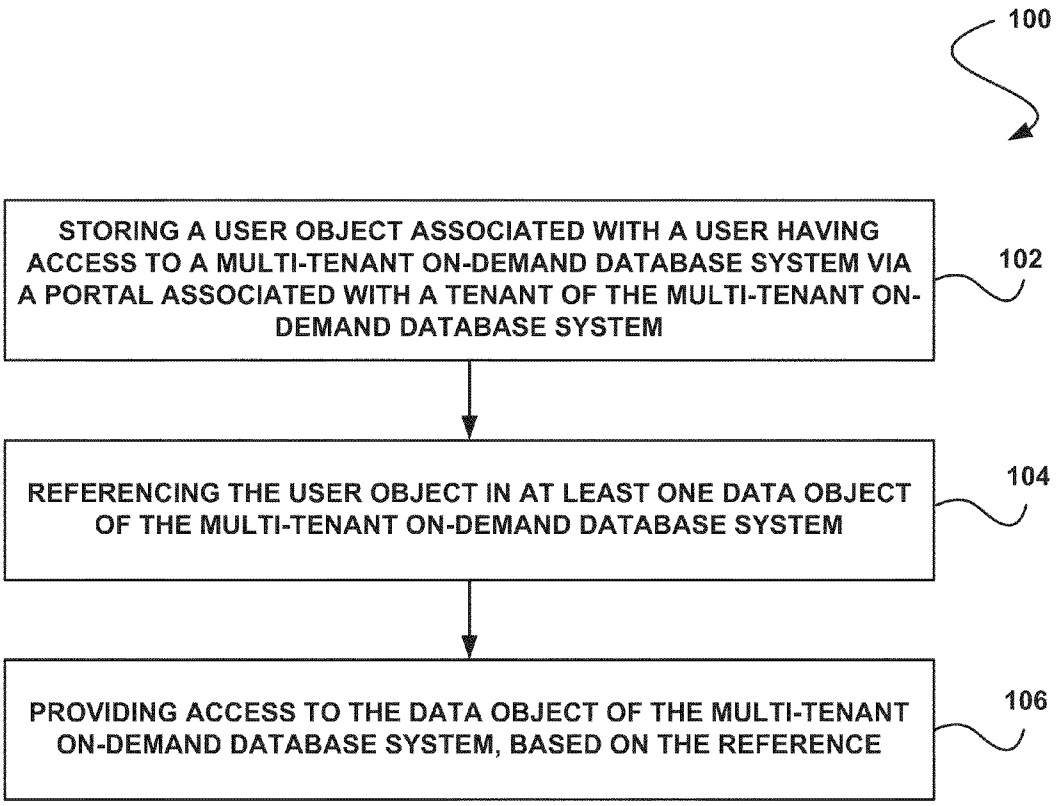

100
STORING A USER OBJECT ASSOCIATED WITH A USER HAVING ACCESS TO A MULTI-TENANT ON-DEMAND DATABASE SYSTEM VIA A PORTAL ASSOCIATED WITH A TENANT OF THE MULTI-TENANT ON-DEMAND DATABASE SYSTEM
102
REFERENCING THE USER OBJECT IN AT LEAST ONE DATA OBJECT OF THE MULTI-TENANT ON-DEMAND DATABASE SYSTEM
104
PROVIDING ACCESS TO THE DATA OBJECT OF THE MULTI-TENANT ON-DEMAND DATABASE SYSTEM, BASED ON THE REFERENCE
106

FIGURE 1

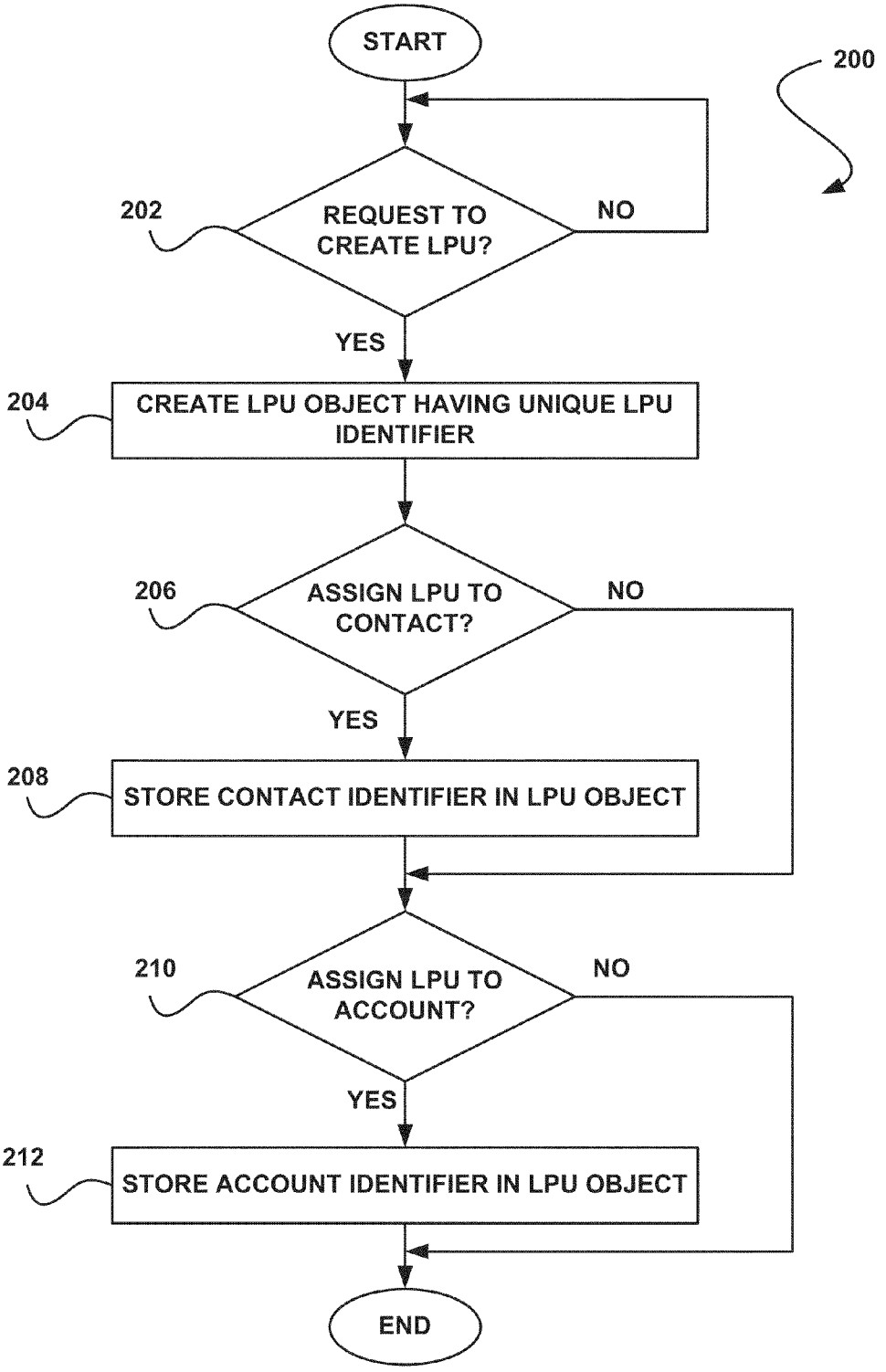

START
200
202
REQUEST TO CREATE LPU?
NO
YES
204
CREATE LPU OBJECT HAVING UNIQUE LPU IDENTIFIER
206
ASSIGN LPU TO CONTACT?
NO
YES
208
STORE CONTACT IDENTIFIER IN LPU OBJECT
210
ASSIGN LPU TO ACCOUNT?
NO
YES
212
STORE ACCOUNT IDENTIFIER IN LPU OBJECT
END

FIGURE 2

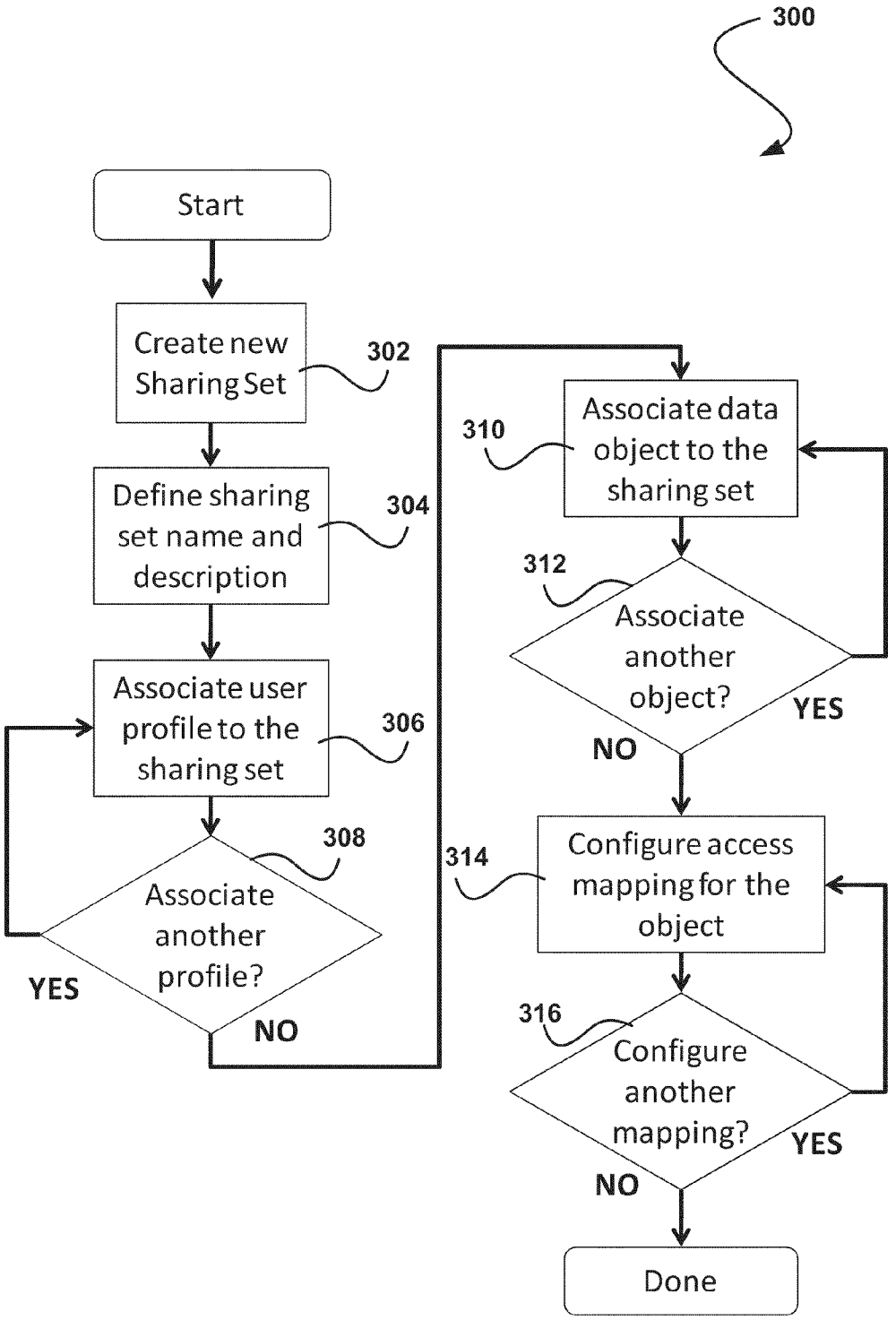

300
Start
Create new Sharing Set
302
Define sharing set name and description
304
Associate user profile to the sharing set
306
Associate another profile?
308
YES
NO
Associate data object to the sharing set
310
Associate another object?
312
YES
NO
Configure access mapping for the object
314
Configure another mapping?
316
YES
NO
Done

FIGURE 3A

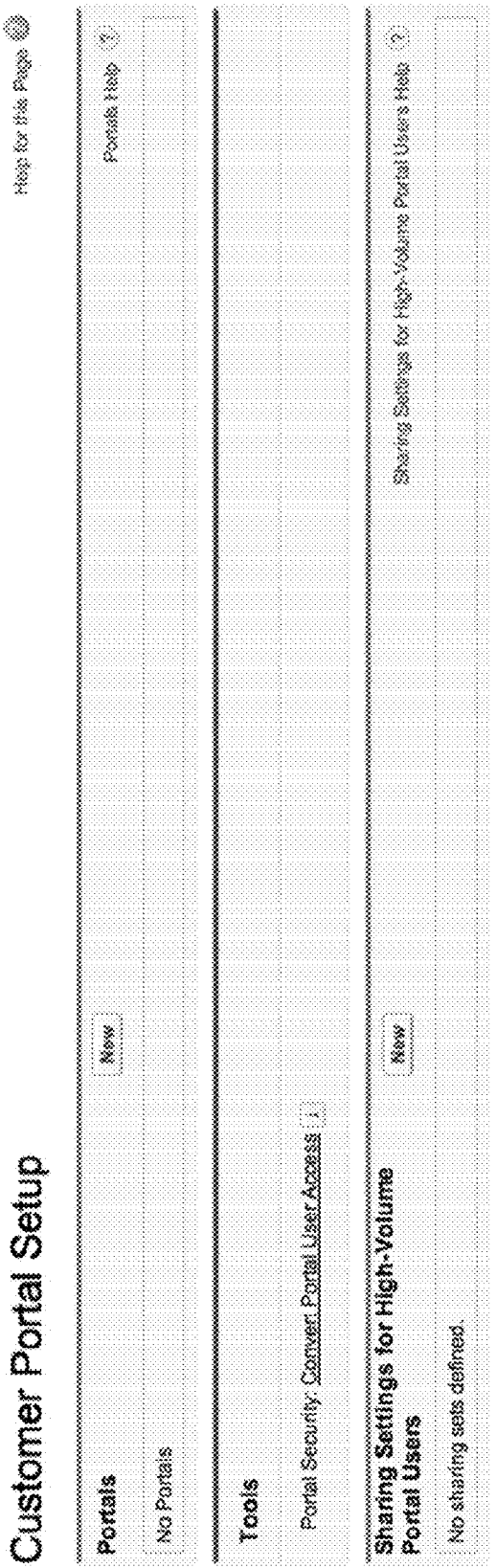

Customer Portal Setup
Help for this Page
Portals
New
Portals Help
No Portals
Tools
Portal Security: Convert Portal User Access
Sharing Settings for High-Volume Portal Users
New
Sharing Settings for High-Volume Portal Users Help
No sharing sets defined.

FIGURE 3B

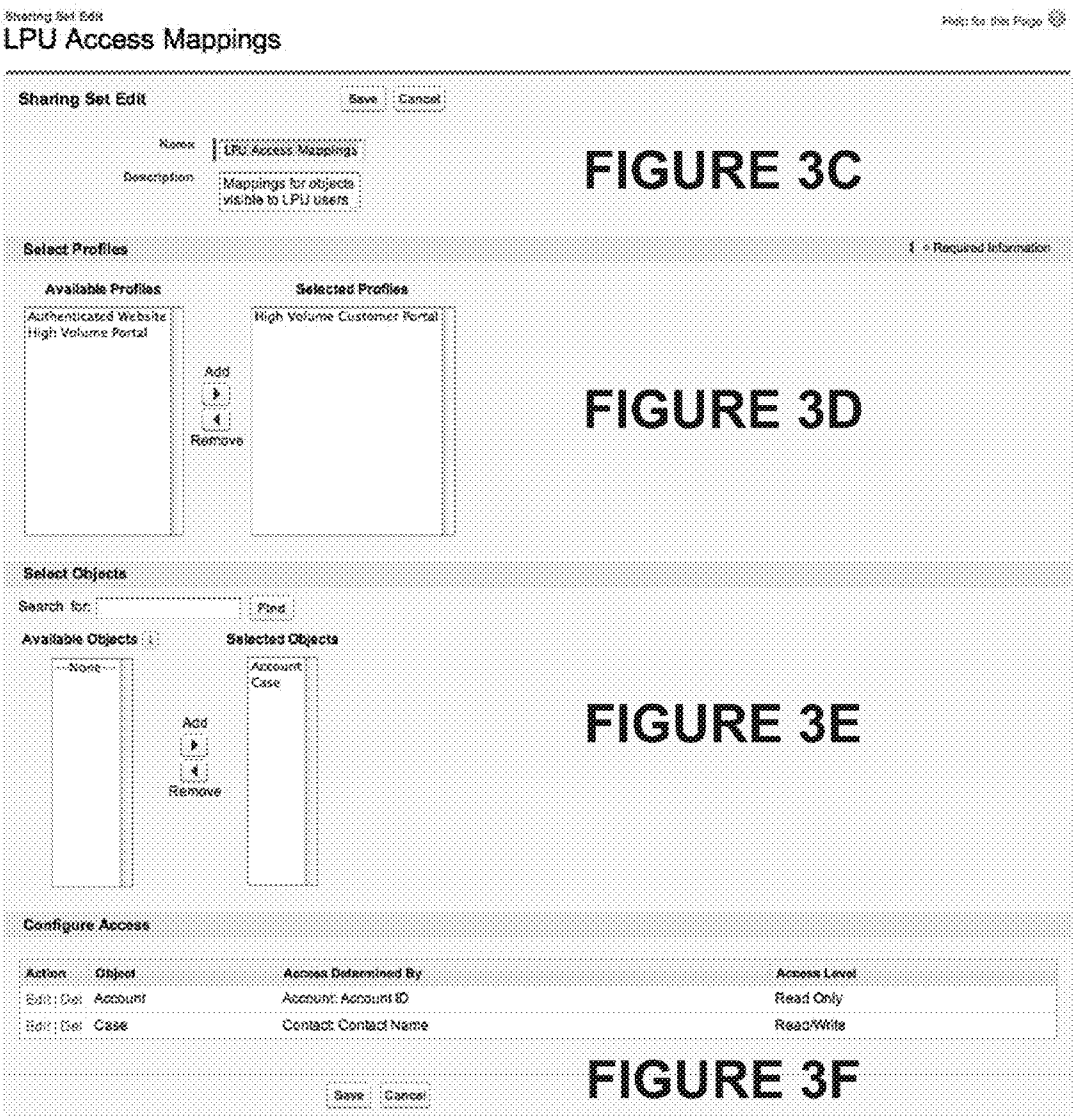


FIGURE 3C

FIGURE 3D

FIGURE 3E

FIGURE 3F

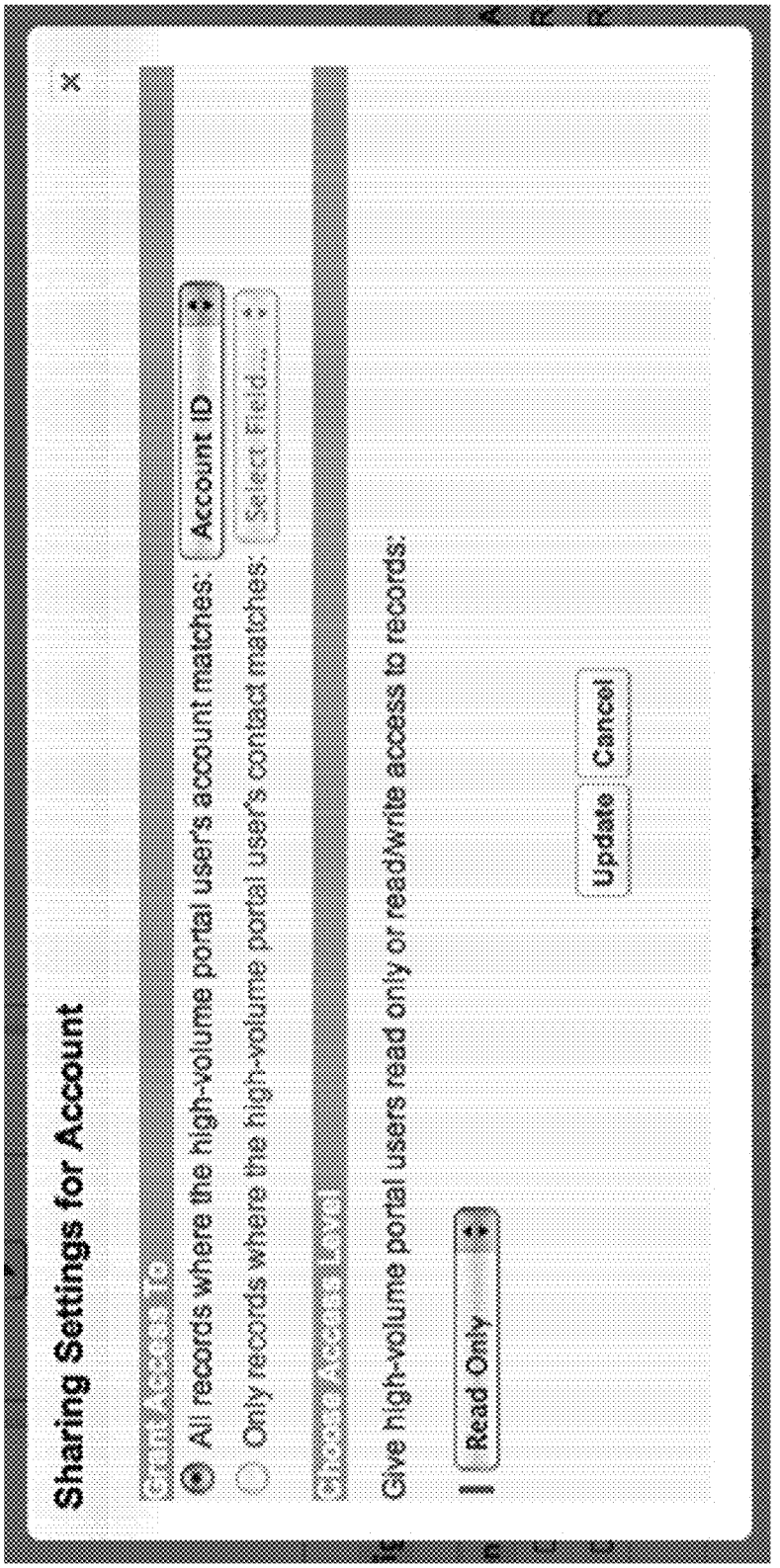

Sharing Settings for Account
Grant Access To
All records where the high-volume portal user's account matches:
Account ID
Only records where the high-volume portal user's contact matches:
Select Field...
Grant Access Level
Give high-volume portal users read only or read/write access to records:
Read Only
Update
Cancel

FIGURE 3G

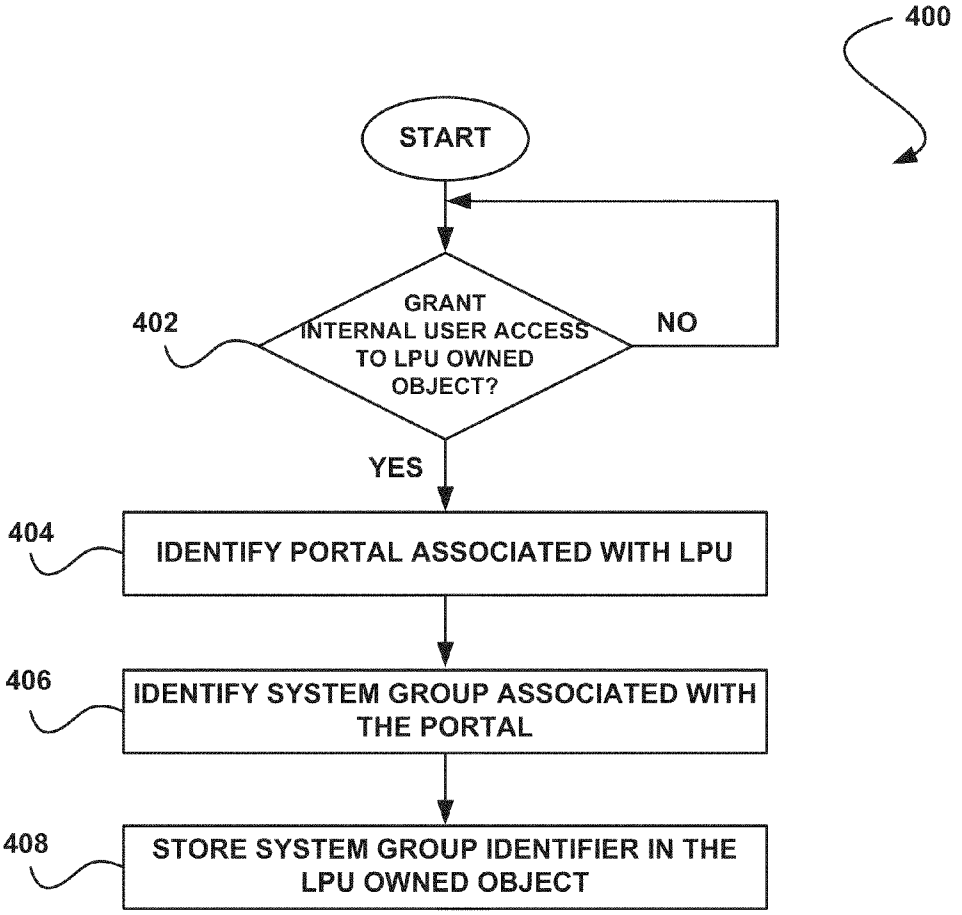

400
START
402
GRANT INTERNAL USER ACCESS TO LPU OWNED OBJECT?
NO
YES
404
IDENTIFY PORTAL ASSOCIATED WITH LPU
406
IDENTIFY SYSTEM GROUP ASSOCIATED WITH THE PORTAL
408
STORE SYSTEM GROUP IDENTIFIER IN THE LPU OWNED OBJECT

FIGURE 4

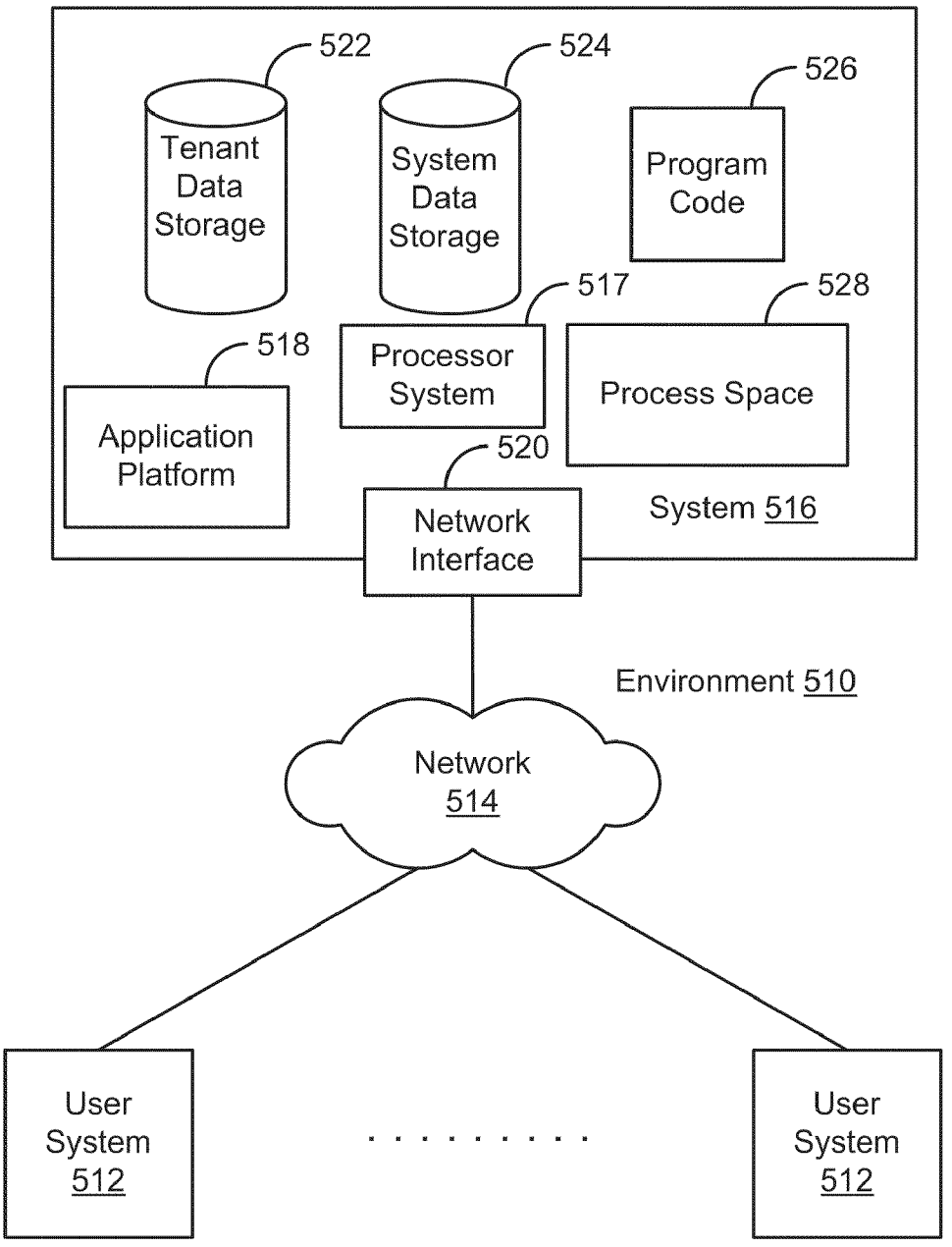

522
524
526
Tenant Data Storage
System Data Storage
Program Code
517
528
518
Processor System
Process Space
Application Platform
520
Network Interface
System 516
Environment 510
Network 514
User System 512
User System 512

FIGURE 5

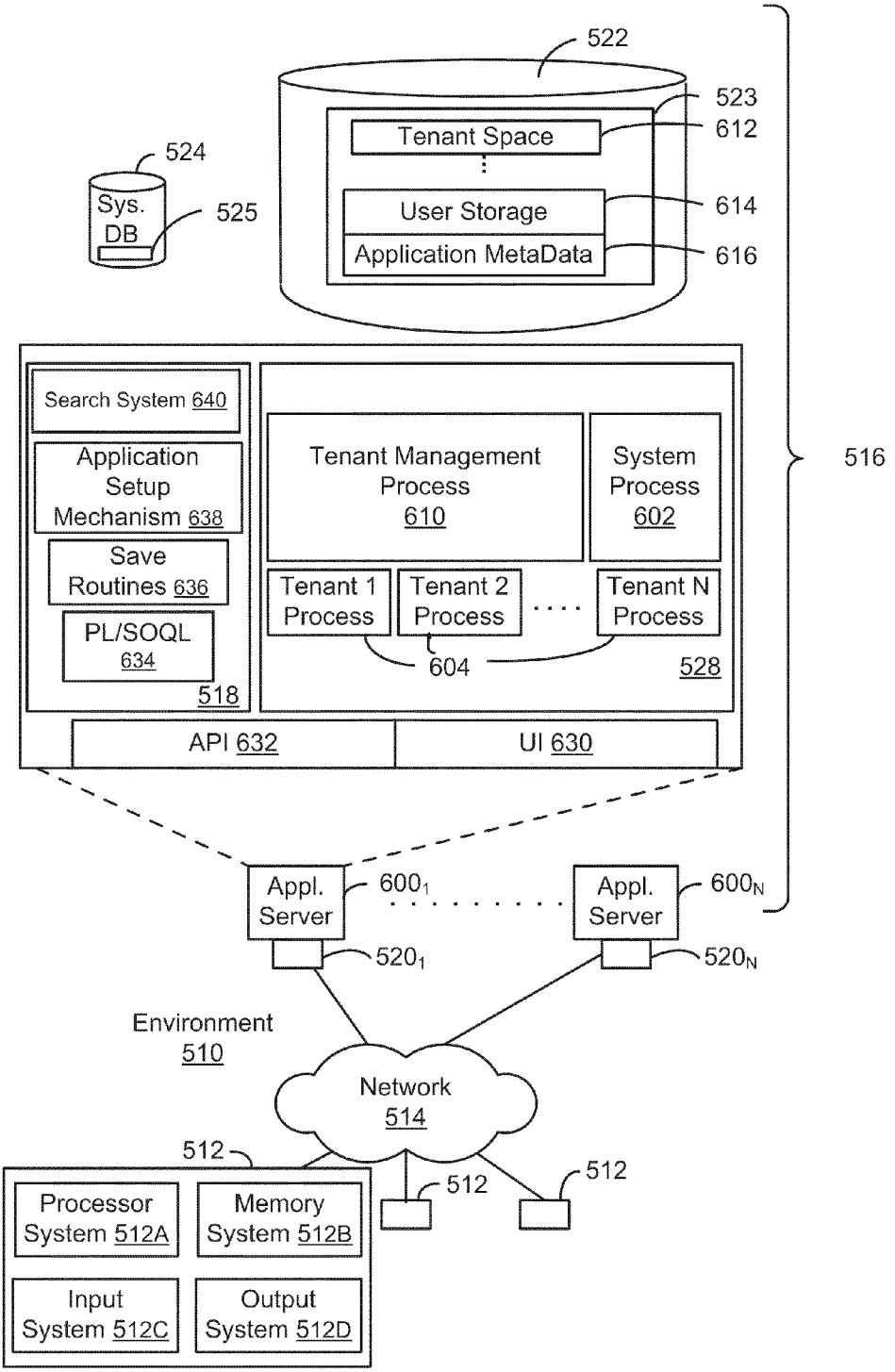

522
523
Tenant Space
612
524
Sys. DB
525
User Storage
614
Application MetaData
616
Search System 640
Application Setup Mechanism 638
Save Routines 636
PL/SOQL 634
518
Tenant Management Process 610
System Process 602
Tenant 1 Process
Tenant 2 Process
Tenant N Process
604
528
516
API 632
UI 630
Appl. Server
600_1
600_N
520_1
520_N
Environment 510
Network 514
512
Processor System 512A
Memory System 512B
Input System 512C
Output System 512D

FIGURE 6

SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR PORTAL USER DATA ACCESS IN A MULTI-TENANT ON-DEMAND DATABASE SYSTEM

CLAIM OF PRIORITY

[0001] This application claims the benefit of U.S. Provisional Patent Application 61/320,152 entitled "METHOD AND SYSTEM FOR GROUP MEMBERSHIP MAINTENANCE AND RECORD ACCESS FOR HIGHLY SCALABLE PORTAL USER SYSTEM," by Wu et al., filed Apr. 1, 2010 (Attorney Docket No, SFC1P112+/306PROV), and U.S. Provisional Patent Application 61/320,188 entitled "METHOD AND SYSTEM FOR AN IMPLICIT SHARING MODEL FOR HIGHLY SCALABLE PORTAL USER SYSTEM," by Wu et al., filed Apr. 1, 2010 (Attorney Docket No. SFC1P113+/307PROV), the entire contents of which are incorporated herein by reference.

COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

FIELD OF THE INVENTION

[0003] One or more implementations relate generally to user access in database environment.

BACKGROUND

[0004] The subject matter discussed in the background section should not be assumed to be prior art merely as a result of its mention in the background section. Similarly, a problem mentioned in the background section or associated with the subject matter of the background section should not be assumed to have been previously recognized in the prior art. The subject matter in the background section merely represents different approaches, which in and of themselves may also be inventions.

[0005] In conventional database system, users access data in the database via an account of the user with the database. The account typically includes login information for verifying the user with the database system, and may further include permissions for indicating data of the database system that the user is allowed to access. Unfortunately, implementations of user accounts in conventional database systems have been limited.

[0006] For example, conventional database systems store only a single type of user account. Accordingly, each user account of the database system is generally configured according to the same types of data, and data access by users having the user account is generally processed in the same manner. This unfortunately does not allow for users of different types which may require different levels of configurations, etc. Just by way of example, users which may not necessarily require the same extent of configurations, permissions, etc, may still be limited to having a type of user account that is the same for other users requiring the configurations, permissions, etc.

[0007] Accordingly, it is desirable to provide techniques enabling the generation of user accounts in a database system having reduced configurations, permissions, etc. to reduce processing requirements of the database system.

BRIEF SUMMARY

[0008] In accordance with embodiments, there are provided mechanisms and methods for portal user data access in a multi-tenant on-demand database system. These mechanisms and methods for portal user data access in a multi-tenant on-demand database system can enable embodiments to provide portal-specific user accounts to the multi-tenant on-demand database system which have reduced configuration requirements than users directly accessing the multi-tenant on-demand database system. The ability of embodiments to provide portal-specific user accounts can reduce processing requirements of the database system.

[0009] In an embodiment and by way of example, a method for portal user data access in a multi-tenant on-demand database system is provided. In use, a user object associated with a user having access to a multi-tenant on-demand database system via a portal associated with the multi-tenant on-demand database system is stored. Additionally, the user object is referenced in at least one data object of the multi-tenant on-demand database system. Furthermore, access to the data object of the multi-tenant on-demand database system is provided, based on the reference.

[0010] While one or more implementations and techniques are described with reference to an embodiment in which portal user data access in a multi-tenant on-demand database system is implemented in a system having an application server providing a front end for an on-demand database service capable of supporting multiple tenants, the one or more implementations and techniques are not limited to multi-tenant databases nor deployment on application servers, Embodiments may be practiced using other database architectures, i.e., ORACLE®, DB2® by IBM and the like without departing from the scope of the embodiments claimed.

[0011] Any of the above embodiments may be used alone or together with one another in any combination. The one or more implementations encompassed within this specification may also include embodiments that are only partially mentioned or alluded to or are not mentioned or alluded to at all in this brief summary or in the abstract. Although various embodiments may have been motivated by various deficiencies with the prior art, which may be discussed or alluded to in one or more places in the specification, the embodiments do not necessarily address any of these deficiencies. In other words, different embodiments may address different deficiencies that may be discussed in the specification. Some embodiments may only partially address some deficiencies or just one deficiency that may be discussed in the specification, and some embodiments may not address any of these deficiencies.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] In the following drawings like reference numbers are used to refer to like elements. Although the following figures depict various examples, the one or more implementations are not limited to the examples depicted in the figures.

[0013] FIG. **1** illustrates a method for portal user data access in a multi-tenant on-demand database system, in accordance with an embodiment;

[0014] FIG. **2** illustrates a method for creating a portal user account, in accordance with an embodiment;

[0015] FIG. **3**A illustrates a method for configuring object access for portal users, in accordance with an embodiment;

[0016] FIG. **3**B illustrates a graphical user interface (GUI) for initiating creation of a sharing set to be used in providing object access to portal users, in accordance with FIG. **3**A;

[0017] FIG. **3**C illustrates a GUI for defining a name and description for a sharing set to be used in providing object access to portal users, in accordance with the method of FIG. **3**A;

[0018] FIG. **3**D illustrates a GUI for associating a portal user profile to the sharing set, in accordance with the method of FIG. **3**A;

[0019] FIG. **3**E illustrates a GUI for associating a data object to the sharing set, in accordance with the method of FIG. **3**A;

[0020] FIG. **3**F illustrates a GUI presenting an access mapping for data objects, in accordance with the method of FIG. **3**A;

[0021] FIG. **3**G illustrates a GUI for configuring an access mapping for data objects, in accordance with the method of FIG. **3**A;

[0022] FIG. **4** illustrates a method for binding individual data records owned by portal users to a share group for the purpose of providing internal users access to the data records, in accordance with an embodiment;

[0023] FIG. **5** illustrates a block diagram of an example of an environment wherein an on-demand database service might be used; and

[0024] FIG. **6** illustrates a block diagram of an embodiment of elements of FIG. **5** and various possible interconnections between these elements.

DETAILED DESCRIPTION

General Overview

[0025] Systems and methods are provided for portal user data access in a multi-tenant on-demand database system.

[0026] As used herein, the term multi-tenant database system refers to those systems in which various elements of hardware and software of the database system may be shared by one or more customers. For example, a given application server may simultaneously process requests for a great number of customers, and a given database table may store rows for a potentially much greater number of customers. As used herein, the term query plan refers to a set of steps used to access information in a database system.

[0027] Next, mechanisms and methods for providing portal user data access in a multi-tenant on-demand database system will be described with reference to example embodiments.

[0028] FIG. **1** illustrates a method **100** for portal user data access in a multi-tenant on-demand database system, in accordance with an embodiment. As shown in operation **102**, a user object associated with a user having access to a multi-tenant on-demand database system via a portal associated with the multi-tenant on-demand database system is stored. In the context of the present description, the portal may include any interface which redirects the user to the multi-tenant on-demand database system. For example, the user may indirectly accesses the multi-tenant on-demand database system via the web interface.

[0029] In one embodiment, the portal may include a web interface which is specific to a tenant (e.g. customer) of the multi-tenant on-demand database system. For example, the multi-tenant on-demand database system may configure the web interface to be customized (e.g. branded) for the tenant. In this way, the user may appear to be accessing a site of the tenant when actually accessing a customized site provided for the tenant by the multi-tenant on-demand database system. In another embodiment, the user may access the multi-tenant on-demand database system via the portal, whereas the tenant may include an internal user which directly accesses the multi-tenant on-demand database system (e.g. via a web interface customized for the multi-tenant on-demand database system).

[0030] As noted above, the multi-tenant on-demand database system stores a user object associated with the user. The user object may include an object used by the user for accessing the portal. For example, the user object may store login information for use in authorizing a login to the portal by the user. Of course, the user object may also store various profile information (e.g. demographic, historical activity, etc.) associated with the user, as an option.

[0031] It should be noted that similarly, the internal user noted above may also have an internal user object stored by the multi-tenant on-demand database system. Such internal user object may be used by the internal user for directly accessing the multi-tenant on-demand database system. For example, the internal user object may store login information for use in authorizing a login to the multi-tenant on-demand database system by the internal user. To this end, in order to access the multi-tenant on-demand database system either directly or indirectly (i.e. via the portal), any type of user may be required to have an associated type of user object stored by the multi-tenant on-demand database system. As an option, the user object may be created upon a registration of the user with the portal/multi-tenant on-demand database system.

[0032] Additionally, as shown in operation **104**, the user object is referenced in at least one data object of the multi-tenant on-demand database system. In the context of the present description, the data object may include any object (e.g. record, etc.) stored by the multi-tenant on-demand database system which is at least potentially capable of being accessed by the user. Thus, the data object may store data that is at least potentially accessible to the user. For example, the data object may be accessible to the user based on permissions for the user to access the data object, as described in more detail below.

[0033] As an option, the data object may be owned (e.g. and therefore managed) by a tenant of the multi-tenant on-demand database system. Just by way of example, the tenant may have created the data object. As another example, the data object may have been assigned to the tenant. Such ownership may optionally be indicated by virtue of a unique identifier of the tenant being stored in a field of the data object used for indicating an owner of the data object. It should be noted that the tenant may include the same or different tenant as the tenant associated with the portal via which the user accesses the multi-tenant on-demand database system.

[0034] In one embodiment, the user object may be directly referenced in the data object. Just by way of example, the user object may store a unique identifier of the user object. The user object may then be referenced by the data object by storing the unique identifier of the user object in the data object. Optionally, the unique identifier of the user object may be stored in a field of the data object which is configured to

store identifiers of user objects to indicate that users of such user objects are allowed to access the data object.

[0035] In another embodiment, the user object may be indirectly referenced in the data object. As an option, a unique identifier of a contact object (representing a contact) may be stored in the user object for relating the user object with the contact object. Further, the user object may be referenced by the data object by storing the unique identifier of the contact object in the data object. As yet another option, a unique identifier of an account object (representing an account) may be stored in the contact object for relating the contact object with the account object, and the user object may be referenced by the data object by storing the unique identifier of the account object in the data object. By using the unique identifier of an account object as described above, access to the data object may be granted to all of the portal users associated with that account object. However, by using the unique identifier of a contact object which may be associated with a unique portal user, as described above, access to the data object may optionally only be granted to the unique portal user.

[0036] Still yet, the user object may be referenced in the data object in response to a manual request for an administrator of the tenant owning the data object. As another option, the user object may be referenced in the data object automatically in response to a determination that the user owns the data object. In yet another embodiment, the user object may be referenced in the data object automatically in response to a determination that the data object is of a public type (e.g. access permissions for the data object are set to public).

[0037] Furthermore, as shown in operation **106**, access to the data object of the multi-tenant on-demand database system is provided, based on the reference. For example, if the data object includes the reference to the user object, then the user may be allowed to access the data object. As another example, if the data object does not include the reference to the user object, then the user may be denied access to the data object.

[0038] To this end, when a user attempts to access the data object (for example via a query for the data object associated with requesting a list, performing a search, running a report, viewing a record, etc.), it may be determined whether the user is authorized to access the data object. In particular, it may be determined whether the user object is referenced by the data object. Access to the data object, for example by returning the data object in a query result, may optionally only be granted when the user object is referenced by the data object. It should be noted that the access to the data object may include reading the data object, writing to the data object, deleting the data object, etc.

[0039] By conditionally providing a user access to data objects based on a reference in the data object of a user object associated with the user, an amount of processing required for determining whether the user is allowed access to the data object may be minimal. For example, use of group membership properties and/or relationship hierarchies for determining data object access may be avoided. In one embodiment, including the user object in a group and referencing the group in the data object for use in granting/denying user access to data objects may be avoided, such that a determination of whether the user associated with the user object is allowed access to the data object (which involves first processing the group to identify all the user objects allowed access to the data object and second comparing the user object of the user requesting access with those identified user objects to determine whether the user is allowed to access the data object) may be avoided.

[0040] To this end, a type of the user object may be different from a type of user object used for internal users of the multi-tenant on-demand database system (i.e. which are configured using the aforementioned group memberships). This may allow the multi-tenant on-demand database system to provide multiple different types of user objects, where the type of user object associated with a user is dependent on the manner in which the user is allowed to access the multi-tenant on-demand database system (i.e. directly or indirectly via the portal). For example, portal users [hereinafter referred to as light portal users (LPUs)] may be associated with user objects granted permission to access data objects in the manner described above with respect to the method **100** of FIG. **1**, whereas internal users may be associated with a different type of user objects storing group membership information, etc. (and thus requiring additional processing for determining permissions to access data objects).

[0041] FIG. **2** illustrates a method **200** for creating a portal user account, in accordance with an embodiment. As an option, the present method **200** may be carried out in the context of the functionality of FIG. **1**. For example, the present method **200** may be carried out utilizing the portal of the multi-tenant on-demand database system described above in the method **100** of FIG. **1**. Of course, however, the method **200** may be carried out in any desired environment. The aforementioned definitions may apply during the present description.

[0042] As shown in decision **202**, it is determined whether a request to create an LPU is received. In one embodiment, the request may be received in response to a user registering with the multi-tenant on-demand database system via a portal. For example, the user may appear to be registering with a service provided by a tenant of the multi-tenant on-demand database system for which the portal is customized by the multi-tenant on-demand database system.

[0043] In another embodiment, the request may be received in response to the tenant registering the user with the multi-tenant on-demand database system on behalf of the user. For example, the user may be an existing customer of the tenant, such that the tenant may register the user with the multi-tenant on-demand database system for use in providing the user with access to data objects of the tenant stored by the multi-tenant on-demand database system. To this end, the request may be directly received by the multi-tenant on-demand database system (e.g. from the tenant) or indirectly received by the multi-tenant on-demand database system (e.g. from the user via the portal).

[0044] If it is determined that a request to create an LPU is not received, the method **200** continues to wait until such a request is received. Once it is determined that a request to create an LPU is received, an LPU object having a unique LPU identifier is created. Note operation **204**. The LPU object may include a predetermined type of object that is configured for registering portal users with the multi-tenant on-demand database system. As noted above, the LPU object has a unique LPU identifier, such as a key for uniquely identifying the LPU object created for the LPU. The LPU object may also include fields for storing login information for the LPU, profile information associated with the LPU, etc.

[0045] In one embodiment, the LPU object may be configured based on information provided by the LPU/tenant. In

another embodiment, the LPU object may be configured automatically using a default profile.

[0046] It is then determined in decision **206** whether the LPU is to be assigned to a contact. The contact may be represented by an existing contact object stored by the multi-tenant on-demand database system or a new contact object not yet stored by the multi-tenant on-demand database system. Thus, the LPU may be assigned to the contact by assigning the LPU object to the contact object (after creation of the contact object if necessary).

[0047] The contact object may optionally store information associated with the LPU which is additional to the information included in the LPU object. Accordingly, the contact object may be a different type of object than the LPU object, and may include different fields than the fields of the LPU object for storing different types of information than the LPU object. Just by way of example, the contact object may store contact information associated with the LPU, such as an email address, a telephone number, an address, etc. In this way, the LPU object may have a one-to-one relationship with the contact object.

[0048] It should be noted that the determination of whether to create the contact object may optionally be based on the registration performed by the LPU or tenant. For example, if the LPU or tenant enters the contact information for the LPU during the registration, then the contact object may be created and the LPU assigned to the associated contact. Of course, as another option, the contact object may be automatically created based on a rule specified by the tenant associated with the portal via which the LPU is to access to the multi-tenant on-demand database system.

[0049] If it is determined that the LPU is to be assigned to a contact, then a contact identifier is stored in the LPU object. Note operation **208**. The contact identifier may include a unique identifier of the contact object associated with the LPU. If, however, it is determined that the LPU is not to be assigned to a contact, or once the LPU is assigned to the contact, it is further determined in decision **210** whether the LPU is to be assigned to an account.

[0050] In the present embodiment, the account may be represented by an existing account object stored by the multi-tenant on-demand database system or a new account object not yet stored by the multi-tenant on-demand database system. Thus, the LPU may be assigned to the account by assigning the LPU object to the account object (after creation of the account object if necessary).

[0051] The account object may optionally store information associated with an account (e.g. portal account) under which the LPU is registering with the multi-tenant on-demand database system. For example, the account may include an account with the tenant of the multi-tenant on-demand database system, and specifically with the portal of the tenant. Accordingly, the account object may store information associated with the LPU which is additional to the information stored by the LPU object and the contact object.

[0052] For example, the account object may be a different type of object than the LPU object and the contact object, and may include different fields than the fields of the LPU object and contact object for storing different types of information than the LPU object and contact object. Just by way of example, the account object may store account information associated with the portal. Since the account may be associated with multiple LPUs, the account object may optionally have a one-to-many relationship with LPU objects.

[0053] It should be noted that the determination of whether to create the account object may optionally be based on the registration performed by the LPU or tenant. For example, if the LPU or tenant enters the account information for the LPU during the registration, then the account object may be created and the LPU assigned to the associated account. Of course, as another option, the account object may be automatically created based on a rule specified by the tenant associated with the portal via which the LPU is to access to the multi-tenant on-demand database system.

[0054] If it is determined that the LPU is to be assigned to an account, then an account identifier is stored in the LPU object. Note operation **212**. The account identifier may include a unique identifier of the account object associated with the LPU. If however, it is determined that the LPU is not to be assigned to an account, or once the LPU is assigned to the account, the method **200** terminates. In this way, an LPU object may be created for an LPU and optionally assigned to a contact object and/or account object, during registration of the LPU with the multi-tenant on-demand database system.

[0055] In an alternative embodiment (not shown), the association of an LPU with one or more objects, such as an account or a contact, could be mandatory for providing access. Thus, for example, the LPU may be required to be assigned to a contact and an account, in an embodiment. For example, the LPU may be directly associated with the contact, and also automatically associated with the parent account of the contact. In yet another embodiment, another type of object present in the system—other than account or contact—could be used by the designers of the system to make associations between LPUs and data objects for the purpose of configuring data access. In yet another embodiment, administrators of the system could themselves create and designate new objects in the system to be used to configure data access to LPUs.

[0056] Table 1 illustrates various examples of LPUs which could be registered using the LPU object. Of course, it should be noted that the examples shown in Table 1 are set forth for illustrative purposes only, and thus should not be construed as limiting in any manner.

TABLE 1

1) users registered with an application store of the multi-tenant on-demand database system;
2) users registered with a service of a tenant of the multi-tenant on-demand database system, e.g. for accessing the multi-tenant on-demand database system for
    a) viewing public objects, such as frequently asked questions, self-support diagnostics, user manuals, etc.
    b) submitting an open issue (e.g. case or ticket) associated with the service
3) users registered for reading, creating, commenting, voting, etc. on data objects (e.g. ideas, discussion topics, etc.) stored by the multi-tenant on-demand database system, etc.
4) users registered for viewing employee information, such as paid time off available, benefits, etc.

[0057] Table 2 illustrates the various internal users who may create and/or utilize the LPU object for various purposes. It should be noted that the portal administrator referenced in Table 2 may include a tenant administrator managing the portal of the tenant. Again, the examples shown in Table 2 are set forth for illustrative purposes only, and thus should not be construed as limiting in any manner.

TABLE 2

Number	"As a . . . " [user type]	" . . . I want to . . . " [required functionality]	" . . . so that I can [use case]"
1	Portal Administrator	create users with the LPU type and profile for my portal	create a portal in which my organization can manage relationships with a very large number of existing and/or potential customers
2	Portal Administrator	provide access for LPUs to records owned by internal users	allow LPUs to participate in internal processes such as ecommerce and customer service, and in community activities such as voting and commenting on Ideas and discussions
3	Portal Administrator	provide access for internal users to records owned by LPUs	establish roles and processes by which internal users can manage the activities of LPUs in sales and support business processes, and can act as moderators and experts in community discussions and activities
4	Portal Administrator	assign LPUs to special portal groups that are mapped to Knowledge Base permissions to Data Dimensions	provide access for those LPUs to the appropriate Knowledge Base Articles
5	Portal Administrator	manage LPUs through an admin UI	maintain appropriate access to portals when users become inactive, need to be reactivated, and/or need to be deleted
6	Portal Developer	create new LPUs and associate them with portals through the Metadata API	automate the creation of new portal users in a scalable way through the portal registration process
7	Portal Developer	assign LPUs to special portal groups mapped to Knowledge Base permissions to Data Dimensions through the Metadata API	automate provision of access for LPUs to the appropriate Knowledge Base Articles in a scalable way through the portal registration process

[0058] FIG. **3**A illustrates a method **300** for configuring object access for portal users, in accordance with an embodiment. As an option, method **300** may be carried out in the context of the functionality of FIGS. **1-2**. For example, the method **300** may be provided to a tenant by the multi-tenant on-demand database system. Of course, however, the method **300** may be carried out in any desired environment. Yet again, the aforementioned definitions may apply during the present description.

[0059] As shown in operation **302**, a new sharing set is created. The sharing set may include any object capable of being utilized for providing object access to portal users. In one embodiment, the sharing set may be created in response to a user selection to create the new sharing set, as shown in FIG. **3**B. Specifically, FIG. **3**B illustrates a GUI for initiating creation of a sharing set to be used in providing object access to portal users.

[0060] In addition, a name and description are defined for the sharing set created in operation **302**. Note operation **304**. The name may include a unique identifier for the sharing set which may be used to subsequently access the sharing set (e.g. via the GUI shown in FIG. **3**B), such as for modifying the sharing set. The description may include text describing the sharing set. The sharing set name and description may be configured by a user, for example, using the GUI shown in FIG. **3**C.

[0061] Further, a user profile is associated to the sharing set, as shown in operation **306**. In the present embodiment, the user profile may include a profile encompassing all portal users (or at least a subset of all portal users). For example, the user profile may be a group of which portal users are members. FIG. **3**D illustrates an exemplary GUI for associating a portal user profile to the sharing set, namely via user selection of the portal user profile.

[0062] It is then determined in decision **308** whether another user profile is to be associated to the sharing set. Accordingly, multiple user profiles may optionally be associated with the sharing set. If it is determined that another user profile is to be associated to the sharing set, the method **300** returns to operation **306**.

[0063] However, if it is determined that another user profile is not to be associated to the sharing set, a data object is associated to the sharing set. Note operation **310**. The data object may include any data object to which access to the portal users of the associated portal user profile is to be granted. FIG. **3**E illustrates a GUI for associating a data object to the sharing set, for example, by allowing user selection of the data object.

[0064] It is then determined in decision **312** whether another data object is to be associated to the sharing set. Thus, multiple data objects may optionally be associated to the sharing set, for allowing the portal users of the associated portal user profile access to such data objects. If it is determined that another data object is to be associated to the sharing set, the method **300** returns to operation **310**.

[0065] In response to a determination that another data object is not to be associated to the sharing set, an access mapping for each data object associated to the sharing set is

configured. Note operation **314**. The access mapping may indicate a manner in which access to each data object associated to the sharing set is determined. For example, access to a particular data object may be granted for portal users of the associated portal user profile being associated with a particular account or a particular contact.

[0066] FIG. **3**F illustrates a GUI presenting an access mapping for data objects. The access mapping is presented by showing for each data object associated to the sharing set, a manner in which access to that data object is determined (e.g. via a particular account or a particular contact), and a type of access (access level) granted to that data object. FIG. **3**G illustrates a GUI for configuring an access mapping for data objects. As shown in FIG. **3**G, a user may select whether access to a data object associated to the sharing set is be granted to portal users of the portal user profile associated to the data set that are associated with (i.e. assigned to) a particular account or a particular contact. As also shown in FIG. **3**G, for the selected option to grant access based on account/contact, the user may further select for which particular account/contact associated portal users of the associated portal user profile are to be granted access to the data object. As further shown in FIG. **3**G, a level of access to be granted to the aforementioned portal users may be configured, where such level may include read-only, read/write, etc.

[0067] To this end, a centralized user interface (e.g. with multiple GUIs) may be provided with the features described in Table 3, for managing LPUs that are members of a portal. It should be noted that the features shown in Table 3 are set forth for illustrative purposes only, and thus should not be construed as limiting in any manner.

TABLE 3

a new page for each portal may allow the portal admin to search through the existing users of the portal, identify a subset of users to manage, and take actions on their membership status
due to the large number of LPUs possibly expected, the minimal requirement for this membership management page may be a list view with full paging and filtering capabilities - the preferred user interface may be task and search based instead of list based and avoid requiring the Portal admin to page through extremely long lists of users
however the Portal Administrator may arrive at the list of users they wish to manage, the UI may also include controls for selecting one or more users and taking action on their membership status, including declaring users inactive, reactivating them if they are already inactive
the ability to create LPUs manually from the UI may be controlled by a new profile permission
Portal Administrators may have the ability to select a "New Light Portal User" button on the list of Portal users
When creating the new user, the Portal Administrator may be restricted to creating users of the Light Portal User type - that is, they may not see any choices for profiles that are not associated with the LPU object type
the Portal Administrator may be able to optionally associate the new LPU with a Portal Account and/or a Contact, in preparation for the visibility model to be configured to LPUs for different objects
the Portal Administrator may be able to inactivate an LPU, and reactive an LPU that is inactive

[0068] FIG. **4** illustrates a method **400** for binding individual data records owned by portal users to a share group for the purpose of providing internal users access to the data records, in accordance with an embodiment. As an option, the present method **400** may be carried out in the context of the functionality of FIGS. **1**-**3**G. For example, the method **400** may be carried out by the multi-tenant on-demand database system. Of course, however, the method **400** may be carried out in any desired environment. The aforementioned definitions may apply during the present description.

[0069] As shown in decision **402**, it is determined whether internal user access is to be granted to an LPU owned object. In the present embodiment, an LPU owned object includes a data object to which an LPU is designated as an owner. As an option, the determination may be made by a tenant of a multi-tenant on-demand database system. For example, the determination may be made by an administrator of the tenant, such that the administrator may determine if any internal users of the tenant (or all internal users of the tenant, and thus the tenant itself) are to be granted access to the LPU owned object.

[0070] In one embodiment, the determination may be based on a tenant administrator requesting to set permissions for an internal user to have access to the LPU owned object. For example, the GUI **340** of FIG. **3**E may be used by the tenant administrator to request that the internal user have access to an LPU owned object, as described above.

[0071] If it is determined that internal user access is not to be granted to an LPU owned object, the method **400** continues to wait for an indication that internal user access is not to be granted to an LPU owned object. Once it is determined that internal user access is to be granted to an LPU owned object, a portal associated with the LPU is determined. Note operation **404**. In the present embodiment, the portal includes a portal provided by the multi-tenant on-demand database system via which the LPU owning the object accessing the multi-tenant on-demand database system. In one embodiment, the portal may be identified in response to a selection via a GUI of the portal by the tenant administrator (e.g. the GUI **310** of FIG. **3**B listing the portals).

[0072] A system group associated with the portal is then identified, as shown in operation **406**. For example, each portal of a tenant may be associated with a single system group (e.g. via assignment of the system group to the portal). Thus, the system group may be associated with the tenant providing the portal. The system group may optionally be represented by a system group object stored by the multi-tenant on-demand database system. The system group may optionally be automatically identified using a reference between an identifier of the portal and an identifier of the system group.

[0073] Furthermore, as shown in operation **408**, a system group identifier is stored in the LPU owned object. The system group identifier may include a unique identifier of the system group object, for example. By storing the system group identifier in the owned object, all members of the system group (e.g. as configured via the GUI **340** of FIG. **3**E) may be granted access to the LPU owned object. For example, in conjunction with the identification of the system group/portal (operations **404**-**406**), the LPU owned object may also be identified, such that the system group may be assigned to the LPU owned object for specifically granting the members of the system group access to the LPU owned object. In an alternative embodiment, the system group identifier may not necessarily be stored directly in the LPU owned object, but instead may be stored in a separate table referenced by the LPU owned object.

[0074] For increased granularity in granting access to only specific members of the system group, a user interface for internal users to the LPU owned record may be built by the tenant associated with the portal. The user interface may allow filtering of which internal users or groups of internal users are allowed access to the LPU object by referring to data on the LPU owned object itself.

[0075] As noted above, internal users may be granted access to LPU owned objects through a system group associated with a portal. If the internal users are members of an access group of a portal, they will have full access to all the data owned by the LPUs belonging to that portal. This may be implemented by writing share rows to a corresponding share table, and at run time doing a join between the share table and group membership table.

[0076] In such a system, objects may be related to each other in parent-child relationships. For example, one object (the "parent" object) may be associated with 1-n objects of another object type (the "child" objects). When an internal user has access to a child record, the internal user may also gain implicit read access to the parent record. For example, if an internal user has access to a child case, he also has at least read access to the parent account of the case, which may be implemented by writing a share row with full access in a case share table, and writing a share row with read access in an account share table.

[0077] An object—such as an account—may be the parent object of other objects that don't have their own sharing model, such as a contract or contact. In this case, the same access rules and settings that apply to the parent object also determine whether a particular user will have access to any of its child objects. Furthermore, when such a child object is owned by an LPU, and made available to internal users of the system through the system group associated with the portal, a share record may be written in the associated share table of the parent object. This share record may designate that all members of the system group associated with the portal will have "full" access to all child objects of the parent object which are owned by LPU users associated with that portal. Full access would allow these internal users to perform any operation on these objects, including reading, editing, deleting and changing the owner of each object. In such a case the system would provide only read access to other child objects of the same parent object, which are not owned by LPUs. This allows administrators of the system to limit the amount of data to which internal users have full access.

[0078] In another potential embodiment, the level of access provided to the members of the portal system group could be restricted to a lesser level of permission, such as Read or Edit. In yet another embodiment, the level of access provided by the portal system group could be configurable by an administrative user. In yet another embodiment, the level of access could be configurable by the administrator for each type of object that is shared to internal users through the portal system group.

System Overview

[0079] FIG. **5** illustrates a block diagram of an environment **510** wherein an on-demand database service might be used. Environment **510** may include user systems **512**, network **514**, system **516**, processor system **517**, application platform **518**, network interface **520**, tenant data storage **522**, system data storage **524**, program code **526**, and process space **528**. In other embodiments, environment **510** may not have all of the components listed and/or may have other elements instead of, or in addition to, those listed above.

[0080] Environment **510** is an environment in which an on-demand database service exists. User system **512** may be any machine or system that is used by a user to access a database user system. For example, any of user systems **512** can be a handheld computing device, a mobile phone, a laptop computer, a work station, and/or a network of computing devices. As illustrated in FIG. **5** (and in more detail in FIG. **6**) user systems **512** might interact via a network **514** with an on-demand database service, which is system **516**.

[0081] An on-demand database service, such as system **516**, is a database system that is made available to outside users that do not need to necessarily be concerned with building and/or maintaining the database system, but instead may be available for their use when the users need the database system (e.g., on the demand of the users). Some on-demand database services may store information from one or more tenants stored into tables of a common database image to form a multi-tenant database system (MTS). Accordingly, "on-demand database service **516**" and "system **516**" will be used interchangeably herein. A database image may include one or more database objects. A relational database management system (RDMS) or the equivalent may execute storage and retrieval of information against the database object(s). Application platform **518** may be a framework that allows the applications of system **516** to run, such as the hardware and/or software, e.g., the operating system. In an embodiment, on-demand database service **516** may include an application platform **518** that enables creation, managing and executing one or more applications developed by the provider of the on-demand database service, users accessing the on-demand database service via user systems **512**, or third party application developers accessing the on-demand database service via user systems **512**.

[0082] The users of user systems **512** may differ in their respective capacities, and the capacity of a particular user system **512** might be entirely determined by permissions (permission levels) for the current user. For example, where a salesperson is using a particular user system **512** to interact with system **516**, that user system has the capacities allotted to that salesperson. However, while an administrator is using that user system to interact with system **516**, that user system has the capacities allotted to that administrator. In systems with a hierarchical role model, users at one permission level may have access to applications, data, and database information accessible by a lower permission level user, hut may not have access to certain applications, database information, and data accessible by a user at a higher permission level. Thus, different users will have different capabilities with regard to accessing and modifying application and database information, depending on a user's security or permission level.

[0083] Network **514** is any network or combination of networks of devices that communicate with one another. For example, network **514** can be any one or any combination of a LAN (local area network), WAN (wide area network), telephone network, wireless network, point-to-point network, star network, token ring network, hub network, or other appropriate configuration. As the most common type of computer network in current use is a TCP/IP (Transfer Control Protocol and Internet Protocol) network, such as the global internetwork of networks often referred to as the "Internet" with a capital "I," that network will be used in many of the examples herein. However, it should be understood that the networks that the one or more implementations might use are not so limited, although TCP/IP is a frequently implemented protocol.

[0084] User systems **512** might communicate with system **516** using TCP/IP and, at a higher network level, use other common Internet protocols to communicate, such as HTTP, FTP, AES, WAP, etc. In an example where HTTP is used, user

system **512** might include an HTTP client commonly referred to as a "browser" for sending and receiving HTTP messages to and from an HTTP server at system **516**. Such an HTTP server might be implemented as the sole network interface between system **516** and network **514**, but other techniques might be used as well or instead. In some implementations, the interface between system **516** and network **514** includes load sharing functionality, such as round-robin HTTP request distributors to balance loads and distribute incoming HTTP requests evenly over a plurality of servers. At least as for the users that are accessing that server, each of the plurality of servers has access to the MTS' data; however, other alternative configurations may be used instead.

[0085] In one embodiment, system **516**, shown in FIG. **5**, implements a web-based customer relationship management (CRM) system. For example, in one embodiment, system **516** includes application servers configured to implement and execute CRM software applications as well as provide related data, code, forms, webpages and other information to and from user systems **512** and to store to, and retrieve from, a database system related data, objects, and Webpage content. With a multi-tenant system, data for multiple tenants may be stored in the same physical database object, however, tenant data typically is arranged so that data of one tenant is kept logically separate from that of other tenants so that one tenant does not have access to another tenant's data, unless such data is expressly shared. In certain embodiments, system **516** implements applications other than, or in addition to, a CRM application. For example, system **516** may provide tenant access to multiple hosted (standard and custom) applications, including a CRM application. User (or third party developer) applications, which may or may not include CRM, may be supported by the application platform **518**, which manages creation, storage of the applications into one or more database objects and executing of the applications in a virtual machine in the process space of the system **516**.

[0086] One arrangement for elements of system **516** is shown in FIG. **5**, including a network interface **520**, application platform **518**, tenant data storage **522** for tenant data **523**, system data storage **524** for system data **525** accessible to system **516** and possibly multiple tenants, program code **526** for implementing various functions of system **516**, and a process space **528** for executing MTS system processes and tenant-specific processes, such as running applications as part of an application hosting service. Additional processes that may execute on system **516** include database indexing processes.

[0087] Several elements in the system shown in FIG. **5** include conventional, well-known elements that are explained only briefly here. For example, each user system **512** could include a desktop personal computer, workstation, laptop, PDA, cell phone, or any wireless access protocol (WAP) enabled device or any other computing device capable of interfacing directly or indirectly to the Internet or other network connection. User system **512** typically runs an HTTP client, e.g., a browsing program, such as Microsoft's Internet Explorer browser, Netscape's Navigator browser, Opera's browser, or a WAP-enabled browser in the case of a cell phone, PDA or other wireless device, or the like, allowing a user (e.g., subscriber of the multi-tenant database system) of user system **512** to access, process and view information, pages and applications available to it from system **516** over network **514**. Each user system **512** also typically includes one or more user interface devices, such as a keyboard, a mouse, trackball, touch pad, touch screen, pen or the like, for interacting with a graphical user interface (GUI) provided by the browser on a display (e.g., a monitor screen, LCD display, etc.) conjunction with pages, forms, applications and other information provided by system **516** or other systems or servers. For example, the user interface device can be used to access data and applications hosted by system **516**, and to perform searches on stored data, and otherwise allow a user to interact with various GUI pages that may be presented to a user. As discussed above, embodiments are suitable for use with the Internet, which refers to a specific global internetwork of networks. However, it should be understood that other networks can be used instead of the Internet, such as an intranet, an extranet, a virtual private network (VPN), a non-TCP/IP based network, any LAN or WAN or the like.

[0088] According to one embodiment, each user system **512** and all of its components are operator configurable using applications, such as a browser, including computer code run using a central processing unit such as an Intel Pentium® processor or the like. Similarly, system **516** (and additional instances of an MTS, where more than one is present) and all of their components might be operator configurable using application(s) including computer code to run using a central processing unit such as processor system **517**, which may include an Intel Pentium® processor or the like, and/or multiple processor units. A computer program product embodiment includes a machine-readable storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the processes of the embodiments described herein. Computer code for operating and configuring system **516** to intercommunicate and to process webpages, applications and other data and media content as described herein are preferably downloaded and stored on a hard disk, but the entire program code, or portions thereof, may also be stored in any other volatile or non-volatile memory medium or device as is well known, such as a ROM or RAM, or provided on any media capable of storing program code, such as any type of rotating media including floppy disks, optical discs, digital versatile disk (DVD), compact disk (CD), microdrive, and magneto-optical disks, and magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data. Additionally, the entire program code, or portions thereof, may be transmitted and downloaded from a software source over a transmission medium, e.g., over the Internet, or from another server, as is well known, or transmitted over any other conventional network connection as is well known (e.g., extranet, VPN. LAN, etc.) using any communication medium and protocols (e.g., TCP/IP, HTTP, HTTPS, Ethernet, etc.) as are well known. It will also be appreciated that computer code for implementing embodiments can be implemented in any programming language that can be executed on a client system and/or server or server system such as, for example, C, C++, HTML, any other markup language, Java™, JavaScript, ActiveX, any other scripting language, such as VBScript, and many other programming languages as are well known may be used. (Java™ is a trademark of Sun Microsystems, Inc.).

[0089] According to one embodiment, each system **516** is configured to provide webpages, forms, applications, data and media content to user (client) systems **512** to support the access by user systems **512** as tenants of system **516**. As such, system **516** provides security mechanisms to keep each tenant's data separate unless the data is shared. If more than one

MTS is used, they may be located in close proximity to one another (e.g., in a server farm located in a single building or campus), or they may be distributed at locations remote from one another (e.g., one or more servers located in city A and one or more servers located in city B). As used herein, each MTS could include one or more logically and/or physically connected servers distributed locally or across one or more geographic locations. Additionally, the term "server" is meant to include a computer system, including processing hardware and process space(s), and an associated storage system and database application (e.g., OODBMS or RDBMS) as is well known in the art. It should also be understood that "server system" and "server" are often used interchangeably herein. Similarly, the database object described herein can be implemented as single databases, a distributed database, a collection of distributed databases, a database with redundant online or offline backups or other redundancies, etc., and might include a distributed database or storage network and associated processing intelligence.

[0090] FIG. **6** also illustrates environment **510**. However, in FIG. **6** elements of system **516** and various interconnections in an embodiment are further illustrated. FIG. **6** shows that user system **512** may include processor system **512**A, memory system **512**B, input system **512**C, and output system **512**D. FIG. **6** shows network **514** and system **516**. FIG. **6** also shows that system **516** may include tenant data storage **522**, tenant data **523**, system data storage **524**, system data **525**, User Interface (UI) **630**, Application Program Interface (API) **632**, PL/SOQL **634**, save routines **636**, application setup mechanism **638**, applications servers $\mathbf{600}_1$-$\mathbf{600}_N$, system process space **602**, tenant process spaces **604**, tenant management process space **610**, tenant storage area **612**, user storage **614**, and application metadata **616**. In other embodiments, environment **510** may not have the same elements as those listed above and/or may have other elements instead of, or in addition to, those listed above.

[0091] User system **512**, network **514**, system **516**, tenant data storage **522**, and system data storage **524** were discussed above in FIG. **5**. Regarding user system **512**, processor system **512**A may be any combination of one or more processors. Memory system **512**B may be any combination of one or more memory devices, short term, and/or long term memory. Input system **512**C may be any combination of input devices, such as one or more keyboards, mice, trackballs, scanners, cameras, and/or interfaces to networks. Output system **512**D may be any combination of output devices, such as one or more monitors, printers, and/or interfaces to networks. As shown by FIG. **6**, system **516** may include a network interface **520** (of FIG. **5**) implemented as a set of HTTP application servers **600**, an application platform **518**, tenant data storage **522**, and system data storage **524**. Also shown is system process space **602**, including individual tenant process spaces **604** and a tenant management process space **610**. Each application server **600** may be configured to tenant data storage **522** and the tenant data **523** therein, and system data storage **524** and the system data **525** therein to serve requests of user systems **512**. The tenant data **523** might be divided into individual tenant storage areas **612**, which can be either a physical arrangement and/or a logical arrangement of data. Within each tenant storage area **612**, user storage **614** and application metadata **616** might be similarly allocated for each user. For example, a copy of a user's most recently used (MRU) items might be stored to user storage **614**. Similarly, a copy of MRU items for an entire organization that is a tenant might be stored to tenant storage area **612**. A UI **630** provides a user interface and an API **632** provides an application programmer interface to system **516** resident processes to users and/or developers at user systems **512**. The tenant data and the system data may be stored in various databases, such as one or more Oracle™ databases.

[0092] Application platform **518** includes an application setup mechanism **638** that supports application developers' creation and management of applications, which may be saved as metadata into tenant data storage **522** by save routines **636** for execution by subscribers as one or more tenant process spaces **604** managed by tenant management process **610** for example. Invocations to such applications may be coded using PL/SOQL **634** that provides a programming language style interface extension to API **632**. A detailed description of some PL/SOQL language embodiments is discussed in commonly owned co-pending U.S. Provisional Patent Application 60/828,192 entitled, PROGRAMMING LANGUAGE METHOD AND SYSTEM FOR EXTENDING APIS TO EXECUTE IN CONJUNCTION WITH DATABASE APIS, by Craig Weissman, filed Oct. 4, 2006, which is incorporated in its entirety herein for all purposes. Invocations to applications may be detected by one or more system processes, which manages retrieving application metadata **616** for the subscriber making the invocation and executing the metadata as an application in a virtual machine.

[0093] Each application server **600** may be communicably coupled to database systems, e.g., having access to system data **525** and tenant data **523**, via a different network connection. For example, one application server $\mathbf{600}_1$ might be coupled via the network **514** (e.g., the Internet), another application server $\mathbf{600}_{N-1}$ might be coupled via a direct network link, and another application server $\mathbf{600}_N$ might be coupled by yet a different network connection. Transfer Control Protocol and Internet Protocol (TCP/IP) are typical protocols for communicating between application servers **600** and the database system. However, it will be apparent to one skilled in the art that other transport protocols may be used to optimize the system depending on the network interconnect used.

[0094] In certain embodiments, each application server **600** is configured to handle requests for any user associated with any organization that is a tenant. Because it is desirable to be able to add and remove application servers from the server pool at any time for any reason, there is preferably no server affinity for a user and/or organization to a specific application server **600**. In one embodiment, therefore, an interface system implementing a load balancing function (e.g., an F5 Big-IP load balancer) is communicably coupled between the application servers **600** and the user systems **512** to distribute requests to the application servers **600**. In one embodiment, the load balancer uses a least connections algorithm to route user requests to the application servers **600**. Other examples of load balancing algorithms, such as round robin and observed response time, also can be used. For example, in certain embodiments, three consecutive requests from the same user could hit three different application servers **600**, and three requests from different users could hit the same application server **600**. In this manner, system **516** is multi-tenant, wherein system **516** handles storage of, and access to, different objects, data and applications across disparate users and organizations.

[0095] As an example of storage, one tenant might be a company that employs a sales force where each salesperson uses system **516** to manage their sales process. Thus, a user

might maintain contact data, leads data, customer follow-up data, performance data, goals and progress data, etc., all applicable to that user's personal sales process (e.g., in tenant data storage **522**). In an example of a MTS arrangement, since all of the data and the applications to access, view, modify, report, transmit, calculate, etc., can be maintained and accessed by a user system having nothing more than network access, the user can manage his or her sales efforts and cycles from any of many different user systems. For example, if a salesperson is visiting a customer and the customer has Internet access in their lobby, the salesperson can obtain critical updates as to that customer while waiting for the customer to arrive in the lobby.

[0096] While each user's data might be separate from other users' data regardless of the employers of each user, some data might be organization-wide data shared or accessible by a plurality of users or all of the users for a given organization that is a tenant. Thus, there might be some data structures managed by system **516** that are allocated at the tenant level while other data structures might be managed at the user level. Because an MTS might support multiple tenants including possible competitors, the MTS should have security protocols that keep data, applications, and application use separate. Also, because many tenants may opt for access to an MTS rather than maintain their own system, redundancy, up-time, and backup are additional functions that may be implemented in the MTS. In addition to user-specific data and tenant specific data, system **516** might also maintain system level data usable by multiple tenants or other data. Such system level data might include industry reports, news, postings, and the like that are sharable among tenants.

[0097] In certain embodiments, user systems **512** (which may be client systems) communicate with application servers **600** to request and update system-level and tenant-level data from system **516** that may require sending one or more queries to tenant data storage **522** and/or system data storage **524**. System **516** (e.g., an application server **600** in system **516**) automatically generates one or more SQL statements (e.g., one or more SQL queries) that are designed to access the desired information. System data storage **524** may generate query plans to access the requested data from the database.

[0098] Each database can generally be viewed as a collection of objects, such as a set of logical tables, containing data fitted into predefined categories. A "table" is one representation of a data object, and may be used herein to simplify the conceptual description of objects and custom objects. It should be understood that "table" and "object" may be used interchangeably herein. Each table generally contains one or more data categories logically arranged as columns or fields in a viewable schema. Each row or record of a table contains an instance of data for each category defined by the fields. For example, a CRM database may include a table that describes a customer with fields for basic contact information such as name, address, phone number, fax number, etc. Another table might describe a purchase order, including fields for information such as customer, product, sale price, date, etc. In some multi-tenant database systems, standard entity tables might be provided for use by all tenants. For CRM database applications, such standard entities might include tables for Account, Contact, Lead, and Opportunity data, each containing pre-defined fields. It should be understood that the word "entity" may also be used interchangeably herein with "object" and "table".

[0099] In some multi-tenant database systems, tenants may be allowed to create and store custom objects, or they may be allowed to customize standard entities or objects, for example by creating custom fields for standard objects, including custom index fields. U.S. patent application Ser. No. 10/817,161, filed Apr. 2, 2004, entitled "Custom Entities and Fields in a Multi-Tenant Database System", and which is hereby incorporated herein by reference, teaches systems and methods for creating custom objects as well as customizing standard objects in a multi-tenant database system. In certain embodiments, for example, all custom entity data rows are stored in a single multi-tenant physical table, which may contain multiple logical tables per organization. It is transparent to customers that their multiple "tables" are in fact stored in one large table or that their data may be stored in the same table as the data of other customers.

[0100] While one or more implementations have been described by way of example and in terms of the specific embodiments, it is to be understood that one or more implementations are not limited to the disclosed embodiments. To the contrary, it is intended to cover various modifications and similar arrangements as would be apparent to those skilled in the art. Therefore, the scope of the appended claims should be accorded the broadest interpretation so as to encompass all such modifications and similar arrangements.

1. A computer program product, comprising a non-transitory computer usable medium having a computer readable program code embodied therein, the computer readable program code adapted to be executed to implement a method for portal user data access in a multi-tenant on-demand database system, the method comprising:

storing a user object associated with a user having access to a multi-tenant on-demand database system via a portal associated with the multi-tenant on-demand database system;

referencing the user object in at least one data object of the multi-tenant on-demand database system; and

providing access to the data object of the multi-tenant on-demand database system, based on the reference, utilizing a processor.

2. The computer program product of claim **1**, wherein the portal includes a web interface specific to a tenant of the multi-tenant on-demand database system.

3. The computer program product of claim **2**, wherein the user indirectly accesses the multi-tenant on-demand database system via the web interface.

4. The computer program product of claim **2**, wherein the tenant includes at least one internal user object for use by an internal user in directly accessing the multi-tenant on-demand database system.

5. The computer program product of claim **1**, wherein the user object is stored by the multi-tenant on-demand database system.

6. The computer program product of claim **1**, wherein the user object stores a unique identifier.

7. The computer program product of claim **7**, wherein the user object is referenced by the data object by storing the unique identifier of the user object in the data object.

8. The computer program product of claim **1**, further comprising storing a unique identifier of a contact object in the user object for relating the user object with the contact object.

9. The computer program product of claim **8**, wherein the user object is referenced by the data object by storing the unique identifier of the contact object in the data object.

10. The computer program product of claim **8**, further comprising storing a unique identifier of an account object in the contact object for relating the contact object with the account object.

11. The computer program product of claim **10**, wherein the user object is referenced by the data object by storing the unique identifier of the account object in the data object.

12. The computer program product of claim **1**, wherein the data object stores data that is at least potentially accessible to the user.

13. The computer program product of claim **1**, wherein the data object is owned by a tenant of the multi-tenant on-demand database system.

14. The computer program product of claim **1**, wherein the access includes reading the data object.

15. The computer program product of claim **1**, further comprising storing another data object owned by the user.

16. The computer program product of claim **15**, further comprising granting a tenant of the multi-tenant on-demand database system access to the other data object owned by the user.

17. The computer program product of claim **16**, wherein the access is granted via a reference stored in the other data object to a system group associated with the tenant.

18. The computer program product of claim **17**, wherein a type of the access granted to the tenant is based on a determination of whether the tenant owns the portal.

19. A method, comprising:

storing a user object associated with a user having access to a multi-tenant on-demand database system via a portal associated with the multi-tenant on-demand database system;

referencing the user object in at least one data object of the multi-tenant on-demand database system; and

providing access to the data object of the multi-tenant on-demand database system, based on the reference.

20. An apparatus, comprising:

a processor for:

storing a user object associated with a user having access to a multi-tenant on-demand database system via a portal associated with the multi-tenant on-demand database system;

referencing the user object in at least one data object of the multi-tenant on-demand database system; and

providing access to the data object of the multi-tenant on-demand database system, based on the reference.

21. A method for transmitting code, comprising:

transmitting code to store a user object associated with a user having access to a multi-tenant on-demand database system via a portal associated with the multi-tenant on-demand database system;

transmitting code to reference the user object in at least one data object of the multi-tenant on-demand database system; and

transmitting code to provide access to the data object of the multi-tenant on-demand database system, based on the reference.

* * * * *