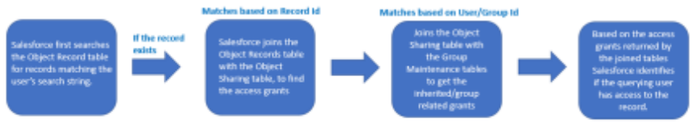
SALESFORCE RECORD SHARING – ACCESS GRANTS & SHARING TABLES

September 26, 2020



Salesforce Record Sharing – Behind the Scenes

In this new digital world, it has become important to properly control who gets to see what data. Data security is a top priority. Users should have access to all the data they need, but should not have access to data they don't need to see.

Record-level access decides which records a user can see for a particular object. There are many ways to share records with users as shown below:

- Organization-wide defaults
- Role hierarchy
- Territory hierarchy
- Sharing rules
- Teams
- Manual sharing
- Programmatic sharing

Because we have so many options for managing record-level access, it's important to understand how Salesforce calculates and grants access at the database level.

Let's dive in and explore Salesforce Security features which will help you configure efficient security models using clicks, not code. We will go through some different options and techniques you can use to understand how Salesforce decides **who gets to see what data**.

For Users/Groups, Sharing objects are used to determine access to a record based on access grants. A good place to start is with access grants.

Access Grants

Access Grants define what access a user or group has to the object's records. It also records which sharing tool is used to provide that access. There are four types of access grants as given below:

- Explicit Grants – When a record is explicitly shared with the user by using Manual Sharing, Sharing rule, Assignment rule, Territory/Programmatic Sharing, etc.
- Group Membership Grants – When a user is a member of the group which has access to the record.
- Inherited Grants – When a user/group inherits the access through a role or territory hierarchy which has access to the record.
- Implicit Grants (built-in sharing) – It gives relational data access to users based on their access to parent/child records.
  A Simplified example of Implicit grant: Users can access(read) parent account record if they have access to its child opportunity, case, or contact record. In a similar way, users can also access child opportunity, case, and contact records if they have access to a parent account record. For more details, please visit Implicit Sharing.

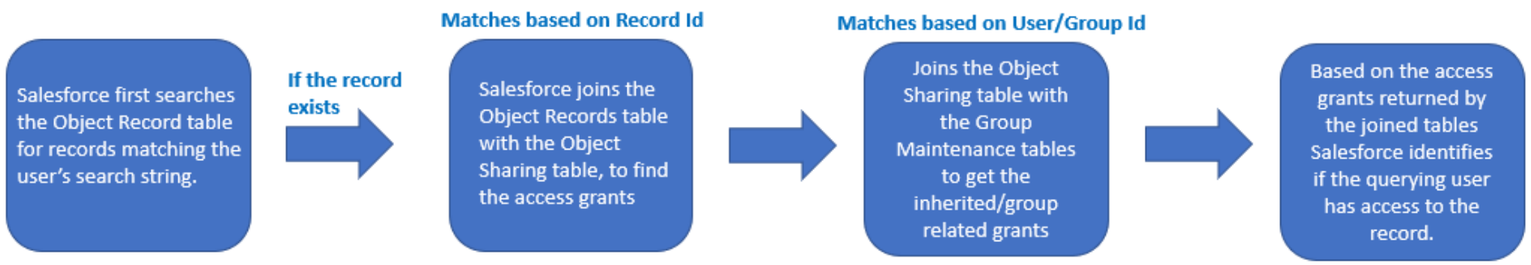
Now that you have an idea about access grants let's understand where are these grants stored?

Sharing & Group Maintenance Tables

Sharing tables store access grants to individuals and groups. Group Maintenance tables store the list of users or groups that belong to each group, indicating group membership. Both types of tables are used to determine a user's access to data when they are accessing a record.

There are three types of tables which are used to determine user's access to a record as demonstrated below:

- Object Record Tables – Stores records of the object.
- Object Sharing Tables – Each object has its own object sharing table which stores implicit & explicit grants. When a record is assigned to a User, Salesforce creates a row in the object's share table for the users with its grant. Each of these rows grant users access to the records.
- Group Maintenance Tables – Generate groups to provide the record access for a set of users, who should have access to the record based on the role hierarchy setup. For example, based on Role hierarchy, managers will be added to all the node groups that are below them.



Salesforce Record Sharing – Behind the Scenes

For users to view the record, the record must exist, and either the Object Sharing table or the Group Maintenance tables must grant access to the querying user.

Both Object Sharing and Group Maintenance tables provide access grants, here is the key difference between those:

- Object Sharing tables store access grants in separate rows called Sharing Rows, each sharing row grants a user or group access to a particular record.
- Group Maintenance tables are more complex because a single group membership or inherited access grant can give several users and groups multiple ways to access a record.

For more details refer Records Sharing & Sharing Tables

Types of Groups

Now let's understand about different types of groups that exist in salesforce and how are these stored in Group Maintenance Tables. There are 2 types of groups as you can see below:

User-defined groups which are created by users and consist of:

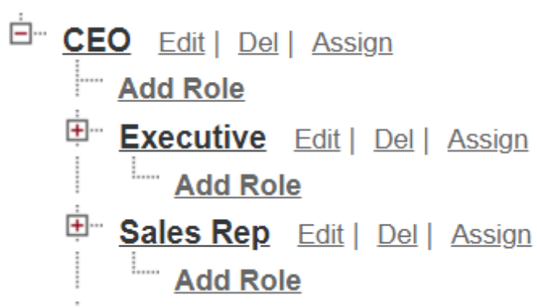
- Public groups
- Private groups

System defined groups which consist of Role groups, RoleAndSubordinates groups, and RoleAndInternalSubordinates groups. RoleAndInternalSubordinates are used if external OWD is enabled. These are based on:

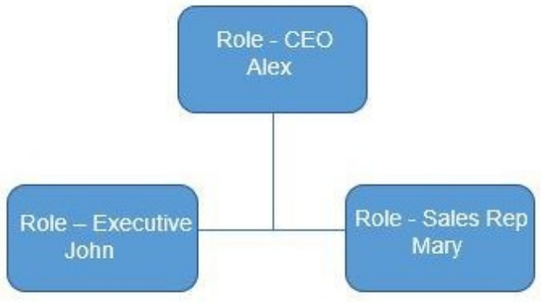
- Role hierarchy
- Territory hierarchy
- Queues

Scenario

Let's understand this with a simple Role hierarchy where Executive & Sales Rep report to the CEO:



As illustrated below, John is an Executive & Mary, who is a Sales Rep, directly report to Alex, who is the CEO of the company.



Object Record Table & Object Sharing Table – Whenever John creates a record of an object, all the records will be stored in the Object's Record table and a sharing row for John as the record owner will be created in the Object Sharing table of that Object.

Example: For all the account records owned by John, Account Object table will have the records stored and Account Sharing table will have access grants records for John with John's User Id, Account Id's & access grants.

Joins of the Object Records table with the Object Sharing table by Record Id-

Whenever John tries to retrieve the Account records from the database, first salesforce tries to see if the record exists in the Account Object table. Once it finds the account record id in Account Record table, then it searches the id in Account Sharing table and retrieves the access grants.

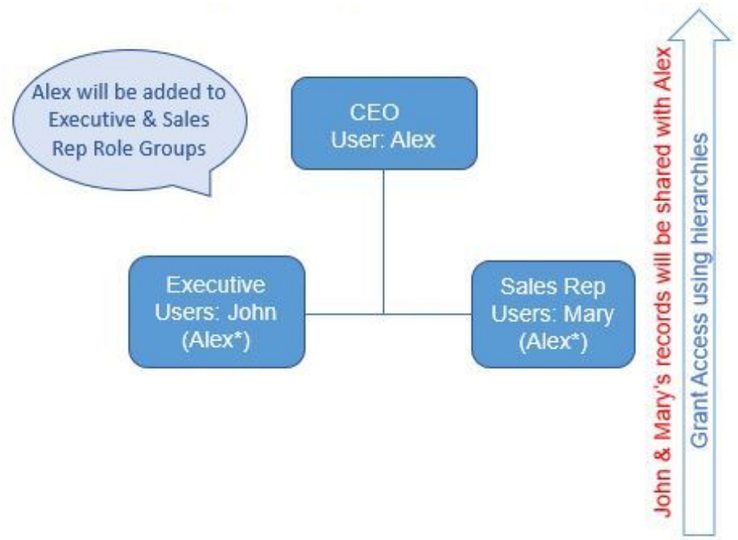
Joins of the Object Sharing table with the Group Maintenance tables by User/Group Id-

Then, Salesforce joins the Object Sharing table with the Group Maintenance tables by using John's Id to fetch the complex access grants such as group membership or inherited access grants which will help to give access to users who are inheriting it from John from the role hierarchy, sharing rule, etc.

For John to see the record from the database, the record must exist in Account Object table, and either the Object Sharing table or the Group Maintenance tables must have the access grants stored for John.

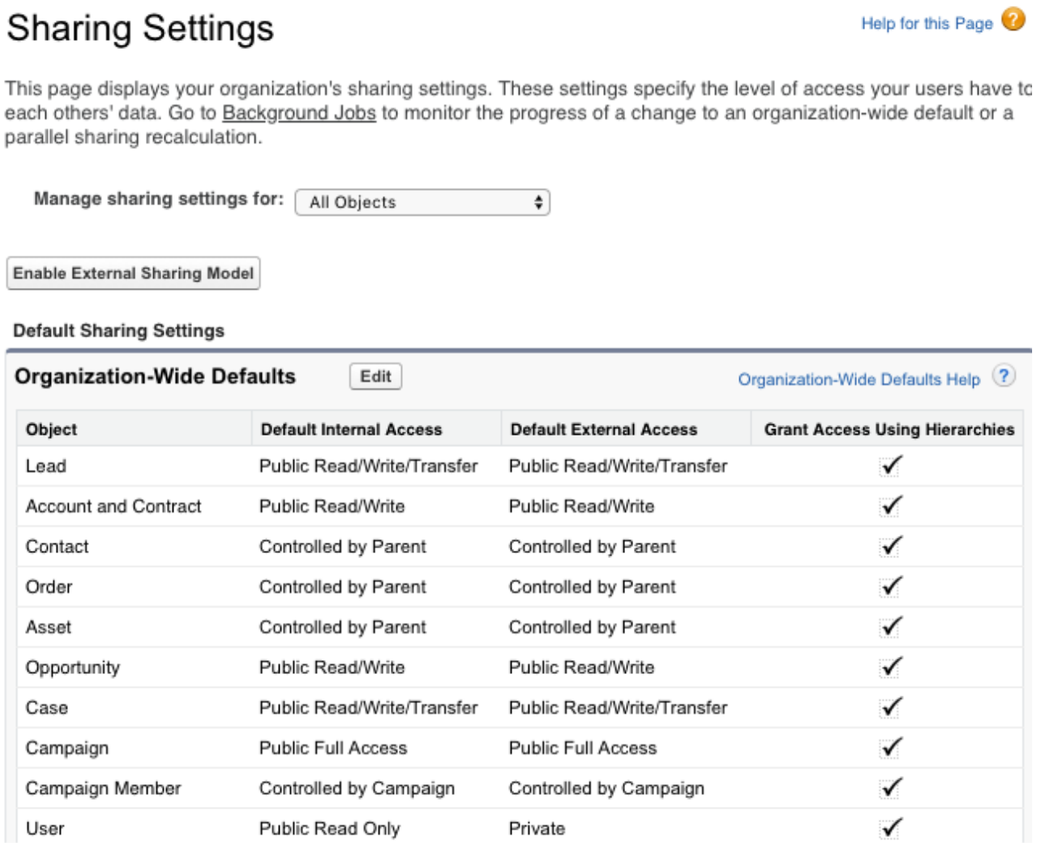
Group Maintenance Table –

Now let's look at the Role groups and RoleAndSubordinates groups created in Group Maintenance tables for nodes in the role hierarchy. Role groups for the hierarchy will be defined as below:



Role hierarchy automatically grants record access to users above the record owner in the hierarchy by " **Grant Access using hierarchies** " checkbox under sharing settings. This option is enabled by default for all objects and can be changed only for Custom Objects.

Grant Access using hierarchies checkbox under sharing settings –



Sharing Settings

Help for this Page

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data. Go to Background Jobs to monitor the progress of a change to an organization-wide default or a parallel sharing recalculation.

Manage sharing settings for: All Objects

Enable External Sharing Model

Default Sharing Settings

Organization-Wide Defaults Edit — Organization-Wide Defaults Help

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	Public Read/Write/Transfer	✓
Account and Contract	Public Read/Write	Public Read/Write	✓
Contact	Controlled by Parent	Controlled by Parent	✓
Order	Controlled by Parent	Controlled by Parent	✓
Asset	Controlled by Parent	Controlled by Parent	✓
Opportunity	Public Read/Write	Public Read/Write	✓
Case	Public Read/Write/Transfer	Public Read/Write/Transfer	✓
Campaign	Public Full Access	Public Full Access	✓
Campaign Member	Controlled by Campaign	Controlled by Campaign	✓
User	Public Read Only	Private	✓

Alex inherits the access from John & Mary as they are below John in the hierarchy so that's the reason Alex is added to John & Mary's Role Groups.

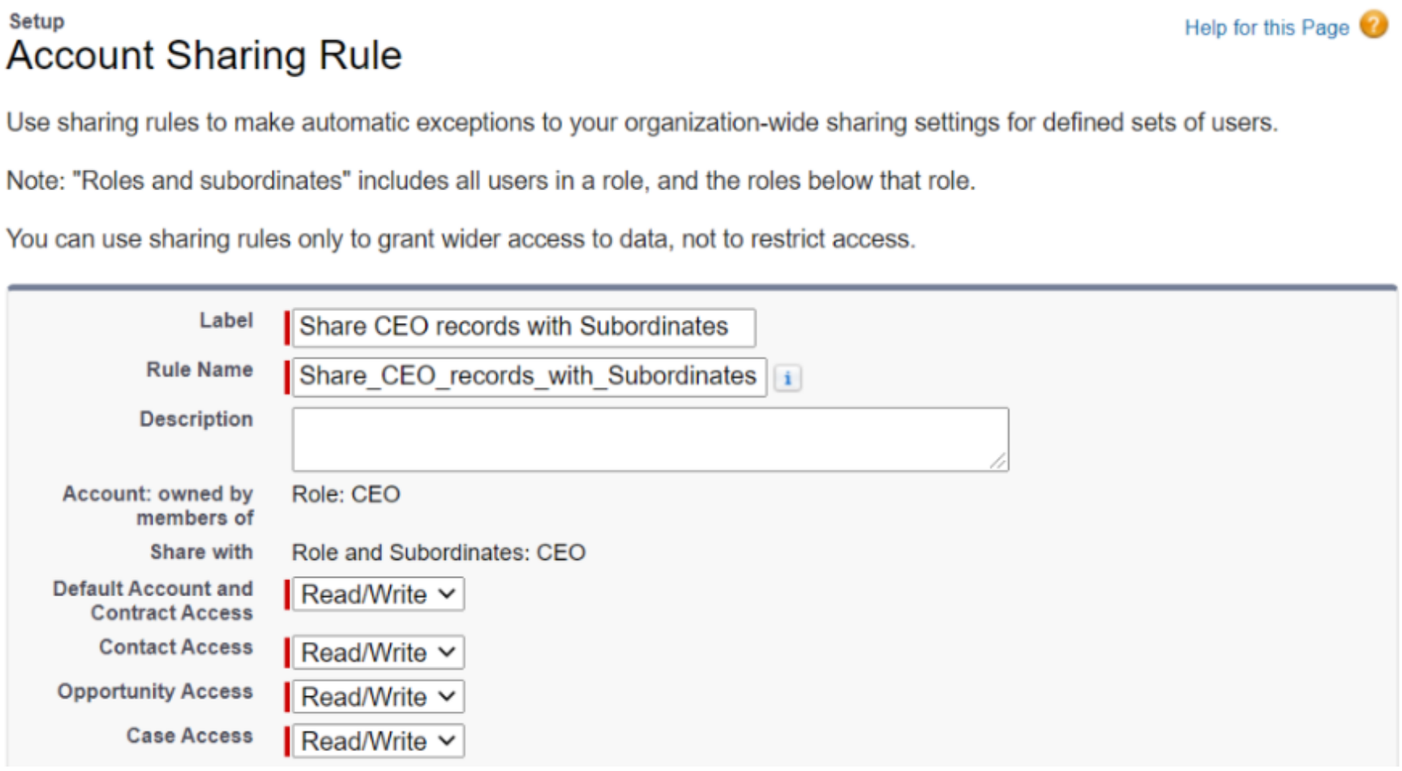
Group Maintenance Tables with Role Groups –

Users	Groups
Alex	CEO Role Group
John	Executive Role Group
Mary	Sales Rep Role Group

Users	Groups
Alex	CEO Role Group
John	Executive Role Group
Mary	Sales Rep Role Group
Alex	**Executive Role Group**
Alex	**Sales Rep Role Group**

Now let's move to RoleAndInternalSubordinates groups. These groups are used whenever the records are shared with subordinates by using sharing rules which share the records with roles and subordinates.

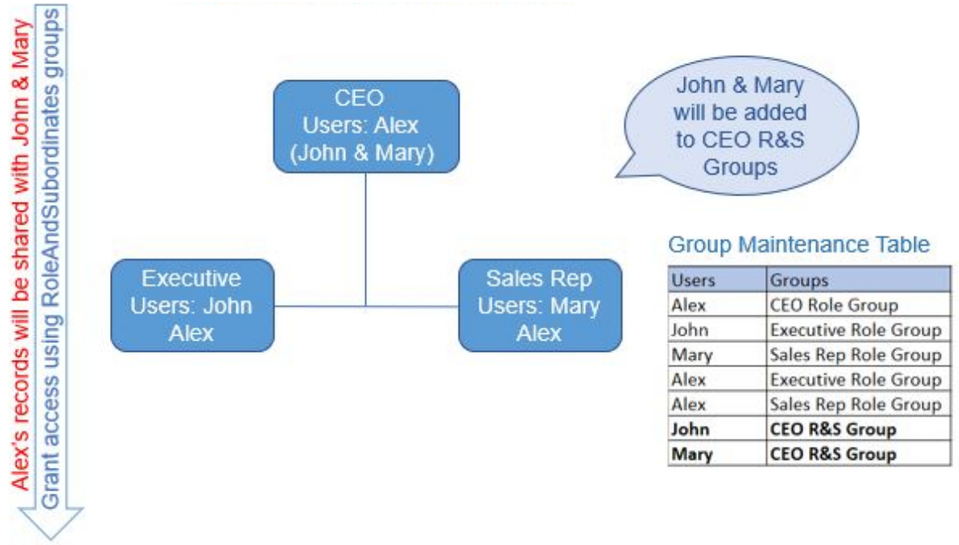
Let's create an Account Sharing Rule which grants CEO's records access to their subordinates:



When you create a sharing rule for an object and share records with

subordinates, Salesforce creates Role & Internal Subordinates groups and adds all the Subordinates to the group. Here in our scenario, all the subordinates of CEO Role will be added to the CEO's Role & Subordinate group. John & Mary will be added to Alex's Role & Subordinate group.

Group Maintenance Table with Role & Subordinate groups –



Group Maintenance Table

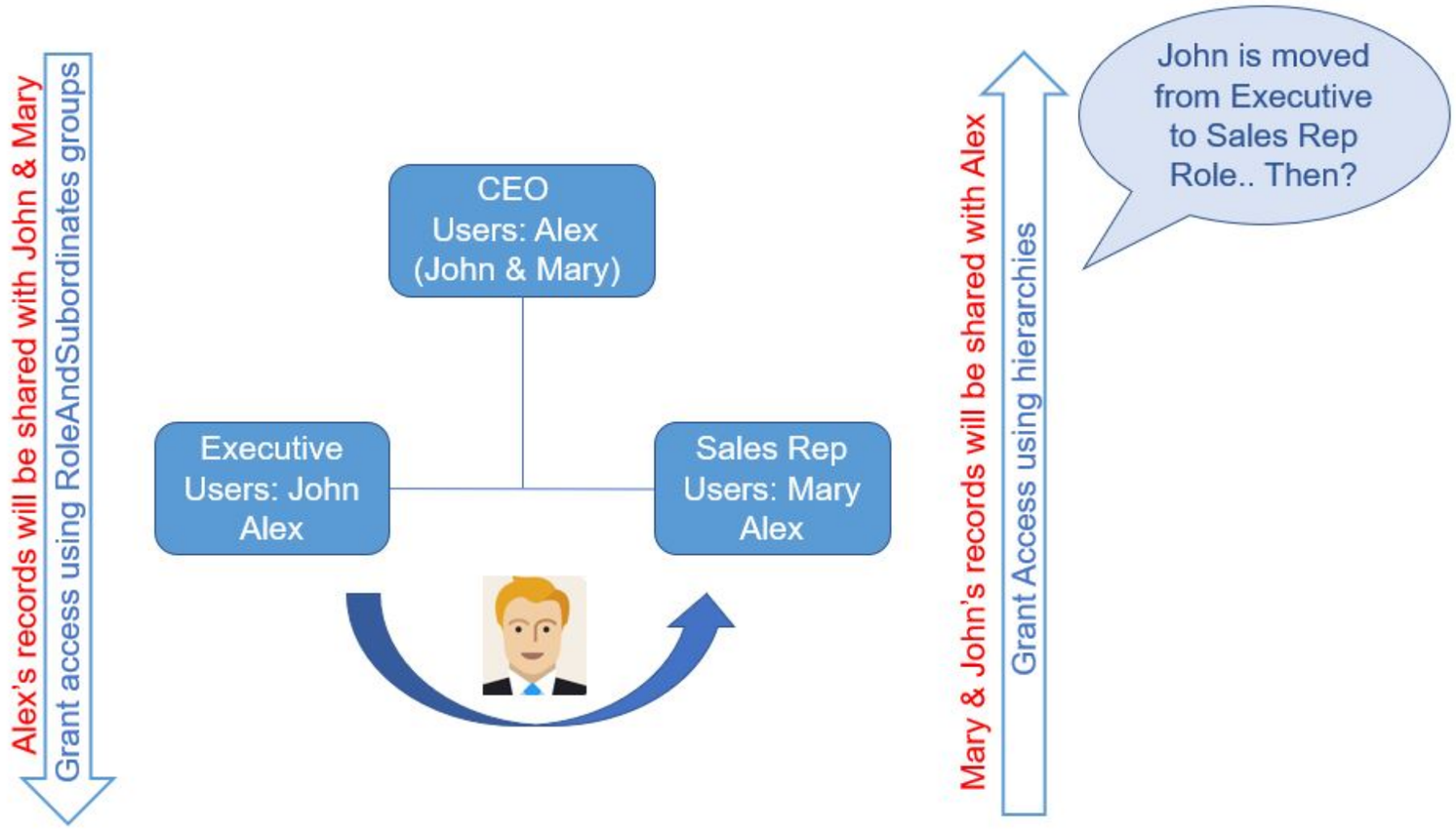
Users	Groups
Alex	CEO Role Group
John	Executive Role Group
Mary	Sales Rep Role Group
Alex	Executive Role Group
Alex	Sales Rep Role Group
John	**CEO R&S Group**
Mary	**CEO R&S Group**

By reading the Role or RoleAndSubordinates groups, Salesforce quickly finds out the indirect members who are part of Role groups and RoleAndSubordinates groups and share the records accordingly. Since John & Mary are added to Alex's RoleAndSubordinates group, Alex's records will be shared with John & Mary. Now, what happens when a user is moved from one role A to role B ?

What happens when the user's role changes

John moves from Executive to Sales Rep role – When a user/administrator takes what looks like a simple action, such as changing the role of a user, there are a lot of checks being performed to determine what the user should see with the new role changes and what should be restricted.

Role Change for a User –



Salesforce recalculates sharing whenever configuration changes occur and performs below actions:

- Adds share records for the objects where access is more permissive with new role
- Removes existing shares where the new settings are more restrictive
- Managers in the branch above the User's old role who were inheriting access will also lose the access. CEO(Alex) will lose the access to the records which they inherited from Executive(John), since John is no more an Executive.
- Add share records for Managers in the branch who will get access based on User's new role

Here when John moves from Executive to Sales Rep role, new share records will be added for John in Sales Rep role. Access to all the records, which john inherited as part of Executive will be removed. All of these take time and resources which in turn affects performance for large data volumes.

Hope this was helpful for you to get good understanding on how access grants are stored in sharing tables, whenever records are shared with users. You can utilize these security features to design your security model to give the right users the right access to the right records. Feel free to share your thoughts.