**Levels of Data Access**

You can control which users have access to which data in your whole org, a specific object, a specific field, or an individual record.

**Organization**

For your whole org, you can maintain a list of authorized users, set password policies, and limit logins to certain hours and locations.

**Objects**  
Access to object-level data is the simplest thing to control. By setting permissions on a particular type of object, you can prevent a group of users from creating, viewing, editing, or deleting any records of that object. For example, you can use object permissions to ensure that interviewers can view positions and job applications but not edit or delete them.  
  
**Fields**  
You can restrict access to certain fields, even if a user has access to the object. For example, you can make the salary field in a position object invisible to interviewers but visible to hiring managers and recruiters.  
  
**Records**You can allow particular users to view an object, but then restrict the individual object records they're allowed to see. For example, an interviewer can see and edit her own reviews, but not the reviews of other interviewers. You can manage record-level access in these four ways.

* **Organization-wide defaults** specify the default level of access users have to each others’ records. You use org-wide sharing settings to lock down your data to the most restrictive level, and then use the other record-level security and sharing tools to selectively give access to other users.
* **Role hierarchies** give access for users higher in the hierarchy to all records owned by users below them in the hierarchy. Role hierarchies don’t have to match your organization chart exactly. Instead, each role in the hierarchy should represent a level of data access that a user or group of users needs.
* **Sharing rules** are automatic exceptions to organization-wide defaults for particular groups of users, so they can get to records they don’t own or can’t normally see. Sharing rules, like role hierarchies, are only used to give additional users access to records. They can’t be stricter than your organization-wide default settings.
* **Manual sharing** allows owners of particular records to share them with other users. Although manual sharing isn’t automated like org-wide sharing settings, role hierarchies, or sharing rules, it can be useful in some situations, such as when a recruiter going on vacation needs to temporarily assign ownership of a job application to someone else.

## Set Password Policy

You can configure several settings to ensure that your users’ passwords are strong and secure.

**Password policies**

Set password and login policies, such as specifying an amount of time before all users’ passwords expire and the level of complexity required for passwords.

**User password expiration**

Expire the passwords for all the users in your org, except for users with “Password Never Expires” permission.

**User password resets**

Reset the password for specified users.

**Login attempts and lockout periods**

If a user is locked out due to too many failed login attempts, you can unlock the person’s access.

**Use Profiles to Restrict Access**

Each user has a single profile that controls which data and features that user has access to. A profile is a collection of settings and permissions. Profile settings determine which data the user can see, and permissions determine what the user can do with that data.

* The settings in a user’s profile determine whether the user can see a particular app, tab, field, or record type.
* The permissions in a user’s profile determine whether the user can create or edit records of a given type, run reports, and customize the app.

Profiles usually match up with a user's job function (for example, system administrator, recruiter, or hiring manager), but you can have profiles for anything that makes sense for your Salesforce org. A profile can be assigned to many users, but a user can have only one profile at a time.

## Use Permission Sets to Grant Access

A permission set is a collection of settings and permissions that give users access to various tools and functions. The settings and permissions in permission sets are also found in profiles, but permission sets extend users’ functional access without changing their profiles.

Permission sets make it easy to grant access to the various apps and custom objects in your org, and to take away access when it’s no longer needed.

Users can have only one profile, but they can have multiple permission sets.

You'll be using permission sets for two general purposes: to grant access to objects or apps, and to grant permissions—temporarily or long term—to specific fields.

## Modify Field-Level Security

Defining field-level security for sensitive fields is the second piece of the security and sharing puzzle, after controlling object-level access.

In some cases, you want users to have access to an object, but limit their access to individual fields in that object. Field-level security settings—or field permissions—control whether a user can see, edit, and delete the value for a particular field on an object. These are the settings that allow us to protect sensitive fields such as a candidate's social security number without having to hide the candidate object.

Unlike page layouts, which only control the visibility of fields on detail and edit pages, field-level security controls the visibility of fields in any part of the app, including related lists, list views, reports, and search results. In fact, to make absolutely sure that a user can't access a particular field, it's important to use the field-level security page for a given object to restrict access to the field. There are simply no other shortcuts that provide the same level of protection for a particular field.

You control record-level access in four ways. They’re listed in order of increasing access. You use org-wide defaults to lock down your data to the most restrictive level, and then use the other record-level security tools to grant access to selected users, as required.

* **Org-wide defaults**specify the default level of access users have to each other’s records.
* **Role hierarchies** ensure managers have access to the same records as their subordinates. Each role in the hierarchy represents a level of data access that a user or group of users needs.
* **Sharing rules** are automatic exceptions to org-wide defaults for particular groups of users, to give them access to records they don’t own or can’t normally see.
* **Manual sharing** lets record owners give read and edit permissions to users who might not have access to the record any other way.

The visibility and access for any type of data is determined by the interaction of the above security controls, based on these key principles.

* A user’s baseline permissions on any object are determined by their profile.
* If the user has any permission sets assigned, these also set the baseline permissions in conjunction with the profile.
* Access to records a user does not own are set first by the org-wide defaults.
* If the org-wide defaults are anything less than **Public Read/Write**, you can open access back up for certain roles using the role hierarchy.
* You can use sharing rules to expand access to additional groups of users.
* Each record owner can manually share individual records with other users by using the Share button on the record.

## Org-Wide Sharing

Org-wide defaults specify the baseline level of access that the most restricted user should have. Use org-wide defaults to lock down your data, and then use the other record-level security and sharing tools (role hierarchies, sharing rules, and manual sharing) to open up the data to users who need it.

Object permissions determine the baseline level of access for all the records in an object. Org-wide defaults modify those permissions for records a user doesn't own. Org-wide sharing settings can be set separately for each type of object.

Org-wide defaults can never grant users more access than they have through their object permission.

you can set the sharing model for that object to one of these settings.

**Private**  
Only the record owner, and users above that role in the hierarchy, can view, edit, and report on those records.  
  
**Public Read Only**

All users can view and report on records, but only the owner, and users above that role in the hierarchy, can edit them.

**Public Read/Write**  
All users can view, edit, and report on all records.

**Controlled by Parent**  
A user can view, edit, or delete a record if she can perform that same action on the record it belongs to.

When the org-wide sharing setting for an object is **Private**or **Public Read Only**, an admin can grant users additional access to records by setting up a role hierarchy or defining sharing rules. Sharing rules can only be used to grant additional access. They cannot be used to restrict access to records beyond what was originally specified with the org-wide sharing defaults.

By default, a role hierarchy automatically grants access to records for users above the record owner in the hierarchy. Setting an object to **Private** makes those records visible *only* to record owners and those above them in the role hierarchy. Use the **Grant Access Using Hierarchies**checkbox to disable access to records to users above the record owner in the hierarchy for custom objects. If you deselect this checkbox for a custom object, only the record owner and users granted access by the org-wide defaults receive access to the records.

Even if **Grant Access Using Hierarchies** is deselected, some users—such as those with the “View All” and “Modify All” object permissions and the “View All Data” and “Modify All Data” system permissions—can still access records they don’t own.

Each sharing rule has three components.

**Share which records?**

You can share records owned by certain users or meeting certain criteria. Criteria-based sharing rules determine what records to share based on field values other than ownership.

**With which users?**

You can define groups of users by role, by territory, or by defining a public group. A public group is an admin-defined grouping of users that can be used to simplify the creation of sharing rules. Depending on the group member types available in your org, public groups can be a combination of:

* individual users
* roles
* roles and subordinates
* territories
* territories and subordinates
* other public groups

**What kind of access?**

You can assign either Read-Only or Read/Write access.

## Define a Public Group

Before creating a sharing rule, it’s important to set up the appropriate public group. A public group is a collection of individual users, other groups, individual roles or territories, and/or roles or territories with their subordinates that all have a function in common. For example, users with the Recruiter profile as well as users in the SW Dev Manager role both review job applications.

Using a public group when defining a sharing rule makes the rule easier to create and, more important, easier to understand later, especially if it's one of many sharing rules that you're trying to maintain in a large organization. Create a public group if you want to define a sharing rule that encompasses more than one or two groups or roles, or any individual.