**🛡️ Data Security in Salesforce**

Data Security in Salesforce manages how data is **protected, shared, and accessed** between users or groups in an organization.  
The **Force.com platform** provides a **layered sharing model**, configurable through the **UI** but enforced at the **API level**, ensuring consistent access control even via integrations.

**Levels of Data Access in Salesforce**

|  |  |  |
| --- | --- | --- |
| **Level** | **Description** | **Controls / Settings** |
| **1.Organization Level** | Controls who can log in and from where. | Authorized users, password policies, login hours, IP restrictions. |
| **2. Object Level** | Determines which objects a user can access and what actions they can perform (CRUD). | Controlled via **Profiles** and **Permission Sets**. |
| **3. Field Level** | Controls visibility/editing of specific fields within an object. | **Field-Level Security (FLS)** in Profiles or Permission Sets. |
| **4. Record Level** | Determines access to specific records within an object. | **OWD**, **Role Hierarchy**, **Sharing Rules**, **Manual Sharing**, **Apex Sharing**. |

**🔐 Control Access to the Organization**

|  |  |
| --- | --- |
| **Method** | **Purpose** |
| **Authorized Users Only** | Only registered users can log in. |
| **Password Policies** | Define password strength and expiration. |
| **IP Range Restrictions** | Limit login by IP range (Org or Profile level). |
| **Login Hours** | Restrict access by time (Profile level only). |

**👥 User Management**

|  |  |
| --- | --- |
| **Component** | **Description** |
| **User Account** | Identifies each Salesforce user. Cannot be deleted (only deactivated or frozen). |
| **Username** | Must be globally unique. |
| **User License** | Determines available features (e.g., Salesforce, Chatter Free). |
| **Profile** | Defines permissions and data visibility based on job role. |
| **Role** | Defines record visibility in role hierarchy (optional, one per user). |
| **Alias** | Short identifier for list views or reports. |

**Deactivate vs Freeze User**

|  |  |  |
| --- | --- | --- |
| **Action** | **Purpose** | **Effect on License** |
| **Deactivate** | Stops login access permanently. | Frees up license. |
| **Freeze** | Temporarily stops login while preparing for deactivation. | License remains in use. |

**🔑 Password Policy Controls**

|  |  |
| --- | --- |
| **Policy** | **Purpose** |
| Password expiration | Set duration before password must be changed. |
| Password resets | Admin can reset passwords for users. |
| Lockout & attempts | Define max login attempts before lockout. |
| “Password Never Expires” | Exception for system or API users. |

**🌐 Restrict Login Access**

|  |  |  |
| --- | --- | --- |
| **Type** | **Level** | **Behavior** |
| **IP Range** | **Org level** | Outside IPs must verify via code (not fully blocked). |
|  | **Profile level** | Login fully denied if IP not in range. |
| **Login Hours** | **Profile level only** | Users can’t log in outside allowed time. Logged-in users lose access after-hours. |

**🔒 Object Level Security (OLS)**

Object Level Security (OLS) is the **simplest and first layer** of data access control in Salesforce.  
It prevents users or groups of users from **creating, viewing, editing, or deleting** any records of an object by setting **object-level permissions**.

**⚙️ Ways to Set Object-Level Permissions**

There are **two main ways** to manage object-level access:

|  |  |
| --- | --- |
| **Method** | **Purpose** |
| **1. Profiles** | Define base-level access for users depending on their job role. |
| **2. Permission Sets** | Extend or add permissions on top of profiles. |

**🧩 Profiles**

A **Profile** is a collection of **settings and permissions** that determine what users can **see** (visibility) and **do** (actions) in Salesforce.

* **Settings** → Define what users *see* (apps, tabs, record types).
* **Permissions** → Define what users *do* (create, edit, delete, run reports, etc.).

**✅ Profiles Control:**

1. Object Permissions (CRUD)
2. Field Permissions (FLS)
3. User Permissions
4. Tab Settings
5. App Settings
6. Apex Class Access
7. Visualforce Page Access
8. Page Layout Assignments
9. Record Types
10. Login Hours
11. Login IP Ranges

**📦 Standard Salesforce Profiles**

|  |  |
| --- | --- |
| **Profile Name** | **Key Permissions** |
| **Standard User** | Read, Edit, Delete on most standard objects. |
| **Read Only** | View-only access; cannot create or edit records. |
| **Marketing User** | Standard User + Campaign & Import Leads access. |
| **Contract Manager** | Standard User + Manage Contracts. |
| **Solution Manager** | Standard User + Manage Solutions. |
| **System Administrator** | Full access + “View All Data” & “Modify All Data.” |

⚠️ **Important Notes:**

* **Standard Profiles** → Object permissions **cannot be edited.**
* **Best Practice:** Clone a standard profile → customize the clone to match your org needs.
* **Every profile** must have **at least one visible app**.
* A **profile can be assigned to multiple users**, but a **user can have only one profile.**

**🧩 Permission Sets**

A **Permission Set** is a group of **settings and permissions** that provide *additional* access to users — **without changing their profile**.

* You need to **grant temporary access** (e.g., to a new feature).
* You want to **avoid creating multiple profiles** for similar roles.

**✅ Permission Sets Control:**

* Object Permissions
* Field Permissions
* User Permissions
* Tab & App Settings
* Apex Class Access
* Visualforce Page Access

**🔹 Profile vs. Permission Set in Salesforce**

|  |  |
| --- | --- |
| **Profile** | **Permission Set** |
| A **Profile** is a collection of settings and permissions that define what a user **can do** in Salesforce (basic access). | A **Permission Set** is a collection of additional permissions that can be **assigned to extend** a user’s access beyond their profile. |
| Controls **base-level access** to the system. | Provides **additional access** without changing the profile. |
| Each user **must have one and only one** profile. | A user can have **multiple permission sets**. |
| Used for defining **minimum required permissions** for a role or job function. | Used to **grant extra permissions** temporarily or permanently for specific tasks or features. |
| Changes to a profile affect **all users** with that profile. | Permission sets can be applied **individually** — changes affect only assigned users. |
| Standard profiles exist by default (e.g., System Administrator, Standard User). | Permission sets must be **created manually**. |
| Can define login hours and IP ranges. | Cannot control login hours or IP ranges. |

**🔐 Field Level Security (FLS)**

**Field Level Security** controls whether a user can **see, edit, or delete** the value of a **specific field** on an object.

* It provides more **granular data protection** than Object-Level Security.
* **Object-Level Security** → Controls access to *entire objects* (e.g., Account, Contact).
* **Field-Level Security** → Controls access to *specific fields* within those objects.

**🧩 FLS Configuration Options**

You can set field-level security in **two main ways:**

|  |  |
| --- | --- |
| **Method** | **Use Case** |
| **1. Through Profiles** | To define base-level access for users. |
| **2. Through Permission Sets** | To extend access for specific users without modifying their profile. |

**👁️‍🗨️ Field Visibility Options**

|  |  |
| --- | --- |
| **Option** | **Meaning** |
| **Visible** | User can view the field’s value. |
| **Read-Only** | User can view but not edit the field. |
| **Hidden** (Unchecked) | User cannot see or access the field anywhere in Salesforce (including reports or API). |

**⚖️ Field-Level Security vs Page Layout**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Field-Level Security (FLS)** | **Page Layout** |
| **Purpose** | Controls *data access* to fields across the platform. | Controls *field visibility* only on record pages. |
| **Scope** | Applies to *everywhere* (reports, list views, API, etc.). | Applies *only* to record detail & edit pages. |
| **Security Enforcement** | Enforced at database level. | UI-based visibility (can be bypassed through reports or API). |
| **Configuration** | Through profiles or permission sets. | Through page layout editor. |

**🔒 Record-Level Security (RLS)**

Record-Level Security determines **which individual records** users can **view, edit, or delete** within an object they already have access to (via Profile or Permission Set).

🧠 **Example:**  
A user may have access to the *Account* object (object-level permission),  
but record-level security decides *which specific Accounts* they can see.

**🧩 Four Ways to Implement Record-Level Security**

**1️⃣Organization-Wide Defaults (OWD)**

OWD defines the **baseline level of access** for all records of an object across the organization. It is the **most restrictive** level of record access.

**OWD Options**

|  |  |
| --- | --- |
| **OWD Setting** | **Description** |
| **Public Read/Write** | All users can view, edit, and report on all records. |
| **Public Read Only** | All users can view and report, but only owners or their managers can edit. |
| **Private** | Only the record owner and users above them in the hierarchy can view, edit, or report. |

**2️⃣ Role Hierarchy**

* A role hierarchy automatically gives **higher-level users** access to all records owned by their subordinates.
* Managers can view, edit, and report on all data owned by users below them in the hierarchy.
* Works automatically for all **standard objects** and can be optionally enabled for **custom objects** via: **“Grant Access Using Hierarchies”** checkbox.
* This option is **always enabled for standard objects** and **can be disabled only for custom objects**.
* Users with **“View All”**, **“Modify All”**, **“View All Data”**, or **“Modify All Data”** permissions override hierarchy restrictions.

**3️⃣ Sharing Rules**

Sharing Rules create **automatic exceptions** to OWD for users who don’t own the record.  
They help open record access to specific roles, public groups, or territories.

⚠️ Sharing rules only **grant additional access** (Read Only or Read/Write) — they **cannot restrict** access.

**Types of Sharing Rules**

|  |  |
| --- | --- |
| **Type** | **Purpose** |
| **Owner-Based** | Shares records owned by specific users, roles, or groups. |
| **Criteria-Based** | Shares records meeting specific field criteria. |

**Before Creating Sharing Rules, Ask:**

* **Which records** should be shared? (based on owner or criteria)
* **With whom** should they be shared? (public groups, roles, etc.)
* **What level of access** is required? (Read-Only or Read/Write)

**4️⃣ Manual Sharing**

Allows **record owners or administrators** to share individual records with specific users, roles, or groups.

**Who Can Manually Share Records**

* Record Owner
* User above the owner in the **role hierarchy**
* User with **Full Access** to the record
* **System Administrator**

**Access Levels**

|  |  |
| --- | --- |
| **Access Level** | **Description** |
| **Full Access** | View, edit, delete, transfer, and share the record (cannot grant full access to others). |
| **Read/Write** | View and edit the record; can add related records, notes, attachments. |
| **Read Only** | View the record; cannot edit or delete it. |
| **Private** | No access to the record. |

🧠 **Note:** The **Share** button appears only when OWD is set to **Private** or **Public Read-Only**.

**👥 Public Groups**

A **Public Group** is a collection of users that can include:

* Individual users
* Roles
* Roles and subordinates
* Other groups

Used primarily in **sharing rules**, **manual sharing**, and **folder/report access** to simplify permission management.