



Home



Library



Profile



Stories



Stats



Following



Alex Rowe



Roger Martin



Dhananjay Garg



The Medium Blog



M.G. Siegler



Anne Bonfert



Krishna Avva



Dr. Benjamin Hardy



Mr. Riyas



Shannon Ashley



More

Our best price of the year.

Get 20% off new memberships for a limited time.

Roles, Profiles, and Permission Sets in Salesforce: Why You Need Eyes, Hands, and Keys



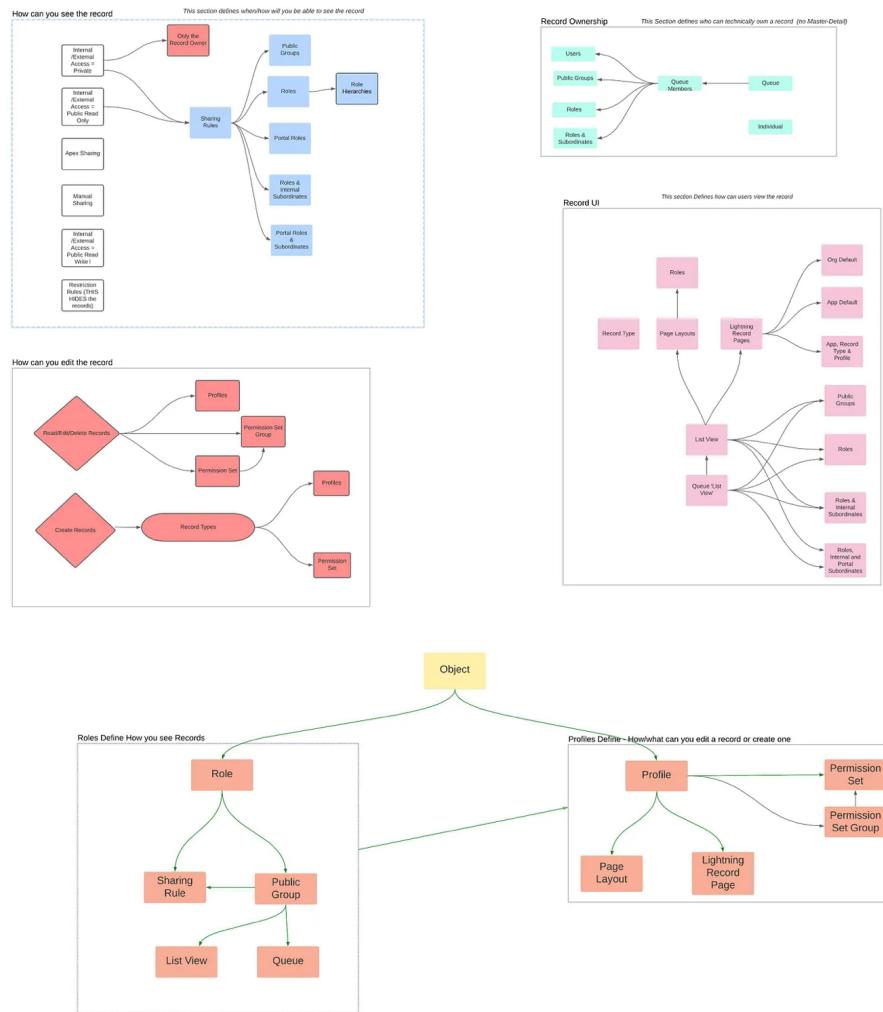
Pranav Nagrecha · 5 min read · Aug 21, 2025



19



Object/Record



Eyes — Hands — Keys

Introduction

If you've ever managed Salesforce, you know this moment:

A VP calls first thing in the morning:

“Why can't I see my team's opportunities? My profile says Edit, but the records are invisible.”

You scramble. You check their profile, their permissions, their role in the org chart. Everything looks fine, and yet they still can't access the records they expect.

Sound familiar?

This isn't just a bug. It's a symptom of the most common and most painful misunderstanding in Salesforce security: **roles ≠ profiles ≠ permission sets**.

The truth is that Salesforce access isn't a single switch. It's a set of **three gates**:

- Roles = Eyes 👀
- Profiles & Permission Sets = Hands 🖐️
- Ownership = Keys 🔑

If you only have eyes, you can look.

If you only have hands, you can grab — but only if you can already see.

And if you don't hold the keys, you don't control who owns the record.

That's the framework. Now let's break it down.

Roles = Eyes

Roles live in the **Role Hierarchy**. They decide whose records you can “see through.”

- Your position in the hierarchy determines which records roll up to you.
- If you’re higher in the tree, you inherit *visibility* into records owned by those below you.
- But here’s the trap: **roles never grant edit rights**. They just open your eyes.

Story:

- Alex is an Account Executive. They own 10 Opportunities.
- Their manager sits above them in the role hierarchy. By default, that manager can see Alex’s Opportunities.
- But unless the manager’s profile or permission set includes **Edit** on Opportunities, they’re just a spectator.

Takeaway:

Roles = the ability to look through the glass.
You're allowed to see, but not touch.

👉 Profiles & Permission Sets = Hands

Profiles and Permission Sets define what you can do once you can see something.

Profiles (the legacy baseline)

Profiles are required. Every user must have one.
They control:

- CRUD (Create, Read, Update, Delete) on each object
- Field-Level Security (FLS) — which fields are visible/editable
- Record Type assignments
- Tab & app visibility
- System permissions like API access
- View All / Modify All on an object (short-circuits sharing)
- View All Data / Modify All Data (org-wide “god mode”)

But here's the critical point: **Salesforce is moving away from Profiles.**

They're meant to be thin ... just the baseline identity (login, default app, time zone). Piling everything into Profiles creates "Franken-profiles" that are impossible to manage.

Permission Sets (the future of access)

Permission Sets are **additive and modular**. They layer extra hands on top of the baseline:

- Grant extra CRUD or FLS
- Add Record Type access
- Assign system permissions
- Can be session-based or conditional
- New features (Muting, PSLs, Conditional Access) **only exist in Permission Sets, not Profiles**

Permission Set Groups (job functions at scale)

Bundles of permission sets that map to real-life roles.

- Example: Case PSG = Case CRUD + Contact Edit + Account View
- Use muted permissions to strip out unnecessary rights.

Story:

- A new Sales Ops hire has a profile with full CRUD on Accounts.
- But the org-wide default (OWD) for Accounts is Private, and their role doesn't put them above anyone who owns Accounts.
- The result: they have hands but no eyes. They could edit Accounts — but they can't see any.

Takeaway:

Profiles and Permission Sets = hands. They let you touch, but only if your eyes (roles/sharing) can already see the record.

👉 Salesforce's future = thin profiles + permission sets/groups for everything else.

🔑 Ownership = Kevs

There's a third gate people forget: ownership.

- Every record has an owner either a user or a queue (for supported objects).
- Ownership matters because:
- Owners often get implicit edit rights.
- Owners can reassign or transfer records if they have “Transfer” permission.
- Sharing models like role hierarchy pivot around who owns the record.
- In master-detail relationships, the child record doesn't even have its own owner , it inherits the keys from the parent.

Story:

- A support case comes in.
- It's auto-assigned to a queue. Anyone in the queue can work on it but only if their profile gives them the right hands.
- Later, a case owner tries to reassign it, but they don't have “Transfer Case” permission. They can see it, they can edit it, but they don't hold the keys.

Takeaway:

Ownership = the keys to the house. Without keys, you can look and touch, but you don't control who really has it.

How They Interlock

The mistake most teams make is treating these as interchangeable. They're not.

- Roles (eyes) control visibility.
 - Profiles & Permission Sets (hands) control capability.
 - Ownership (keys) controls who holds the record.

Think of a locked room:

- Roles decide whether the blinds are open or closed.
 - Profiles & Permission Sets decide if you can rearrange the furniture.
 - Ownership decides who holds the deed to the house.

Miss one, and your user screams “I can’t see it” or “I can’t edit it” even though the other settings look fine.

Classic Misfires (You've Seen These)

VP sees but can't edit

- Role hierarchy gave them eyes into their reps' Opportunities.
- Profile only had Read, no Edit.
- Eyes but no hands.

New hire has CRUD but sees nothing

- Profile = full CRUD on Accounts.
- OWD = Private, no sharing rule.
- Hands but no eyes.

Portal user chaos

- External user gets Record Type access via permission set.
- But External Sharing Model = Private.
- Blindfolded hands.

1. Case queue lockup

- Record assigned to a queue. User is in the queue.

- But profile doesn't allow Edit.
- Keys but no hands.

Blind Spots & Gotchas (Advanced)

- **Modify All vs. Modify All Data** → object-level vs. org-level override.
- **Restriction Rules vs. Scoping Rules** → **Restriction hides records**; Scoping just filters list views/reports.
- **Implicit sharing** → master-detail children vanish when the parent isn't shared.
- **Teams & Territories** → add share rows automatically (Account Team, Opportunity Team, ETM).
- **Manual sharing** → not available for all objects.
- **External Sharing Model** → separate from OWD, controls portal users.
- **Portal Role Hierarchy** → behaves differently than internal hierarchies.

Debugging in the Right Order (The Flow)

When someone yells “I can’t see it,” don’t check everything at once. Go in order:

1.  See (eyes) → OWD → Role → Sharing → Teams/Territories → Restriction → External Sharing.
2.  Edit (hands) → CRUD → FLS → Permission Set (muted?) → Validation Rule → Flow/Trigger → Approval Lock.
3.  Own (keys) → Ownership → Queue vs User → Transfer rights → Assignment rule.

Nine times out of ten, the ghost lives at just one gate.

Checklist: Launching a New Object Safely

- OWD set (usually Private first).
- Role hierarchy impact reviewed.
- CRUD/FLS mapped in Permission Sets.
- Record Types only where process diverges.
- Lightning Page visibility rules documented.
- External Sharing Model defined for portals.
- Restriction Rules applied if needed.
- Ownership model agreed (User vs Queue vs Parent).
- Negative tests done (“should NOT see/edit”).

Visual Reference

Want the **full diagram** of how these pieces interlock?

I mapped it out in Lucid so you can explore and share it with your team:

 [View the full “Eyes, Hands, Keys” diagram on Lucid](#)

Final Thought

Salesforce access isn't a black box. It's three gates.

- Roles = Eyes 👀 — let you look.
- Profiles/Permission Sets = Hands 🖐️ — let you act.
- Ownership = Keys 🔑 — decide who really holds it.

Next time someone screams, don't panic. Just ask:

Do they have eyes? Do they have hands? Do they have keys?

Salesforce

Sales Development

Salesforce Tools

Salesforce Admin



Written by **Pranav Nagrecha**

[Edit profile](#)

51 followers · 43 following

Salesforce

Administrator/Developer/Architect/Busin
ess Analyst/QA

No responses yet



...



Pranav Nagrecha he

What are your thoughts?

More from **Pranav Nagrecha**

 Pranav Nagrecha

Boomi—How to convert Timezones i...

I use Google to the extreme, I know there are smarter...

May 17, 2022



...

 Pranav Nagrecha

How We Built a Real-Time Integration...

Say goodbye to 15-minute polling delays—here's how...

May 14  13



...

 Pranav Nagrecha

Stop Guessing. Start Designing: Order of...

Your save isn't haunted—it's a kitchen line: before-save,...

Aug 26  2



...

 Pranav Nagrecha

Flow Control: Navigating...

Flows are the lifeblood of many business processes....

Sep 25, 2023  9



...

See all from Pranav Nagrecha

Recommended from Medium

 Vaishnavi R

Database. Stateful in Salesforce

If you have ever worked with batch classes in Apex, you...

 Jul 23  2  ...

 Pedro Távora Santos

Why Your Agentforce POC Will Fail: An...

If you're reading this, you've probably sat through the...

 Nov 18  ...

 Musa Ndlala

The Salesforce Sales Pipeline—Where...

In every sales organization, there comes a moment...

 Oct 31

...

 Sai Santhosh Kandibanda

Day 4 -Salesforce All-Star Ranger

Today, we are exploring the module called “Data...

 Nov 22

...

 Jeremy Carmona

How I Use Salesforce Templates to Save...

Third in the “Salesforce, Simplified” series

 Jun 12

12.1K

22

 Uma Srinivas

Salesforce Entitlements

Entitlements in Salesforce represent service contract...



...

[See more recommendations](#)

[Help](#) [Status](#) [About](#) [Careers](#) [Press](#) [Blog](#) [Privacy](#) [Rules](#)
[Terms](#) [Text to speech](#)