

Record Access, Decoded

Every way Salesforce decides who sees a record. The whole sharing & security model, on one page.



Neil Sarkar
CO-FOUNDER, CLIENTELL

THE ONE LAW THAT EXPLAINS THE WHOLE MODEL

Salesforce record access only ever **ADDS**. To take access away you have exactly two record-level levers: lower the **OWD floor**, or build a **Restriction Rule wall**.

Roles, sharing rules, teams, Apex shares: every one of them only stacks more access on top. There is no such thing as a restrictive sharing rule.

GATE 1 · THE WHAT

Object + Field

Can you touch this **kind** of data at all? Profiles, permission sets, FLS. Answers a **WHAT** question.

GATE 2 · THE WHO

Record Access

Can you touch **this specific** record? A ladder that only climbs. you get the highest rung any path reaches. Answers a **WHO** question.

BOTH GATES · AND

The Two Walls

You need **both** gates open to see a value. The only things that pull you back down: **Restriction Rules** (WHO) and **Muting Perm Sets** (WHAT).

1 GATE 1 — OBJECT + FIELD (THE WHAT)

Profiles · Permission Sets · FLS

Profile = baseline (exactly one per user). **Permission Sets + Permission Set Groups** stack on top, always additive.

Muting Permission Set = the WHAT-side wall. Works **only inside a Permission Set Group**. it mutes object, field, AND system permissions granted by that group's other sets. It cannot touch the profile.

View All / Modify All (per object) and **View / Modify All Data** (org-wide) = a **sharing bypass**. they ignore OWD, role hierarchy, and every sharing rule. **But FLS still applies** (you still need the field, or "View All Fields").

FLS = field visibility, **given** object access. Enforced in UI, reports, search, and standard APIs.

Profile note: the "permissions leave profiles" retirement was **postponed, not enforced**. Permission-set-led is still best practice, but profiles keep login hours/IP, default record types & apps, and page-layout assignment.

2 GATE 2 — RECORD ACCESS (THE WHO) = THE LADDER, MAX OF:

Most-permissive grant wins

RUNG	WHAT IT DOES	THE CATCH
1 OWD	Sets the baseline floor every other rung climbs from.	Internal & External are separate settings , and External can never exceed Internal.
2 Role Hierarchy	Rolls access UP to managers ("Grant Access Using Hierarchies").	Forced ON for standard objects. off-able only for custom objects . The usual source of "why can they see this?"
3 Sharing Rules	Owner-based or criteria-based, to public groups / roles / territories.	Open-only . Declarative rules grant Read or Read/Write. they can never restrict.
4 Manual / Teams	"Share" button (only when OWD is below Public R/W). Account/Opp/Case Teams = the same thing with a nicer UI.	Both are deleted on owner change . Never build a process that depends on a manual share surviving.
5 Apex Managed	<code>__Share</code> rows written under a developer-defined sharing reason.	Survives owner change . the durable fix when manual shares keep vanishing.
6 Implicit	Native, always on, can't toggle. runs in both directions (child ↔ account).	Account→Case/Contact/Opp child shares are computed at query time (no stored <code>ImplicitChild</code> rows since Winter '24). Other implicit shares still persist.

OWD VALUES IN FULL (MOST CHEAT SHEETS GET THIS WRONG)

Private · Public Read Only · Public Read/Write · **Public Read/Write/Transfer** (Lead, Case) · **Public Full Access** (Campaign) · **Controlled by Parent** (master-detail detail side, e.g. Contact under Account). Price Books use a separate set: **Use / View Only / No Access**.

GRANT LEVELS

Sharing rules cap at Read/Write, but **manual & Apex shares can grant Full Access** (transfer / delete / reshare) on supported objects.

3 DON'T CONFUSE THESE

The most-botched pair in the whole model

Restriction Rule A WALL

Subtractive. Removes access a user would otherwise have. Enforced on reports, list views, **SOQL / SOSL, search, lookups, related lists, and API**. a real security boundary.

Exception: View All / Modify All users are exempt. (Enterprise+, limited object set.)

Scoping Rule A DEFAULT LENS

Sets what you see **by default** on SOQL, reports, and list views only. The user can switch scope and still reach every record they have access to. **NOT security**.

If a list-view filter can defeat it, it was never security.

4 CODE & AUTOMATION — THE ONE BUG THAT SHIPS TO PROD

Apex · Flow

SHARING-SAFE IS NOT FIELD-SAFE

with / without / inherited sharing control **Gate 2 (records) only. never Gate 1**. Your **with sharing** Apex still happily returns fields the user can't see, unless you enforce FLS yourself.

2026 WATCH On **API v67.0 (Summer '26)**, DB operations default to **user mode** (enforces sharing + FLS + CRUD), no-declaration classes now default to **with sharing**, and **WITH USER_MODE** replaces **WITH SECURITY_ENFORCED**. Legacy classes (≤ v66.0) keep the old system-mode behavior. **Flow** (record-triggered) still defaults to system mode. a silent over-exposure path.

5 THE MONDAY-MORNING LITMUS TESTS

Apply before you build

NEED TO SUBTRACT ACCESS?

Exactly two tools: the OWD floor, or a Restriction Rule wall. Reached for anything else? You're about to over-grant.

REACHING FOR VIEW ALL?

You've already lost. Using a sharing bypass to fix a sharing problem means the design is the problem.

WHO OR WHAT?

Sharing answers WHO. Permissions answer WHAT. Never fix a WHO problem with a WHAT tool.



Clientell AI

Clientell audits your entire access model and **fixes what's broken in minutes**. OWD, roles, sharing rules, FLS, restriction, all in plain English.

app.clientell.ai →



Reports & Dashboards



Flows & Automation



Apex & LWC



Data & Custom Objects



Org Audit & Security



Debug, Test & Deploy