Record-Level Access

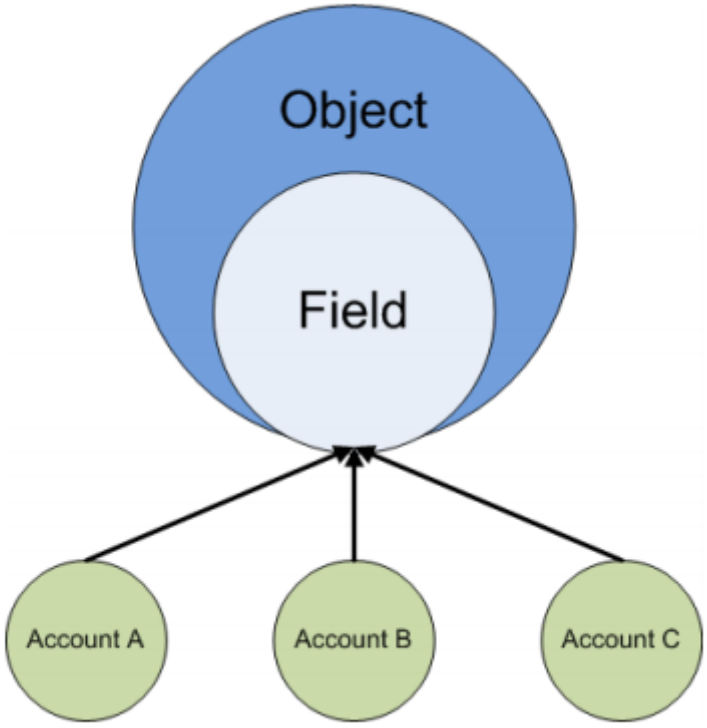
- Record-Level Access

Record-Level Access

To meet your company’s security needs, it’s important to understand what data access means to your users and to you.

Data Access: User’s Perspective:

If you put yourself in your users’ shoes, you won’t necessarily know or care how you’re getting access to records, but you might want to understand what having access means within the context of your organisation. The following graph can help users visualise the different kinds of access that can be configured in Salesforce



For example, if a user has access to an account field, then they have access to both the account field and the account object itself.
However, a specific account record, such as “Account A”, might not be accessible to that user due to additional access control applied via sharing rules or other tools.

Data Access: Architect’s Perspective
As an architect, you must both understand your user’s perspective and know how to grant users only the appropriate level of access to the data that they should be able to access.

From an architect’s perspective, data access in Salesforce falls into two main categories: object-level access, which includes field-level access, and record-level access.

Object-level access determines whether a user has access to a particular object, which fields they can see on that object, and which actions they can perform. You configure object level access on user profiles.

Restricting access
The “Read,” “Create,” “Edit,” and “Delete” object permissions determine which actions a user can perform on any of the object's
records to which they have access. Field-Level Security allows you to prevent certain users from seeing sensitive or confidential
information contained in records they can see.

Opening up access
The “View All” and “Modify All” object permissions give users access to all of an object’s records, regardless of record-level access
settings. Record-level access (called “Sharing” in Salesforce) determines which records a user can see for a particular object, using the following tools:
• Organisation-wide defaults
• Role hierarchy
• Territory hierarchy
• Sharing rules
• Teams
• Manual sharing
• Programmatic sharing

Because you have so many options for managing record-level access — and because some of these options are affected by organisational dependencies — determining which records users can access can quickly become complicated. Additionally, you might also be changing your sharing configuration frequently in response to new business requirements. This can trigger record access changes that ripple through your organisation. These changes have an even greater impact in very large organisations, where it can take some time to recalculate access for a large number of users, and adjust the tables that record their access rights. For these reasons, it’s important to understand how Salesforce calculates and grants access at the database level.

Record Access Calculation
Every time a user attempts to open a record, run a report, access a list view, or search for data using the user interface or API, Salesforce checks the configuration of its record access features to determine which records the user can access. These configurations can be elaborate, especially in large organisations with hundreds of hierarchy nodes, thousands of sharing rules, millions of data

rows, and portals for customers and business partners. Processing such dissimilar data and complex relationships would require far more time than the 300-millisecond Salesforce benchmark for rendering pages. Rather than applying every sharing rule, traversing all hierarchies, and analysing record access inheritance in real time, Salesforce calculates
record access data only when configuration changes occur. The calculated results persist in a way that facilitates rapid scanning and minimises the number of database table joins necessary to determine record access at run time.

Access Grants
When an object has its organisation-wide default set to Private or Public Read Only, Salesforce uses access grants to define how much access a user or group has to that object's records. Each access grant gives a specific user or group access to a specific record. It also records the type of sharing tool — sharing rule, team, etc. — used to provide that access. Salesforce uses four types of access grants: explicit grants, group membership grants, inherited grants, and implicit grants.

Explicit Grants
Salesforce uses explicit grants when records are shared directly to users or groups. Specifically, Salesforce uses explicit grants when:
• A user or a queue becomes the owner of a record.
• A sharing rule shares the record to a personal or public group, a queue, a role, or a territory.
• An assignment rule shares the record to a user or a queue.
• A territory assignment rule shares the record to a territory.
• A user manually shares the record to a user, a personal or public group, a queue, a role, or a territory.
• A user becomes part of a team for an account, opportunity, or case.
• A programmatic customisation shares the record to a user, a personal or public group, a queue, a role, or a territory.

Group Membership Grants
Grants that occur when a user, personal or public group, queue, role, or territory is a member of a group that has explicit access to
the record. For example, if a sharing rule explicitly grants the Strategy group access to the Acme record, and Bob is a member of the Strategy group, Bob's membership in the Strategy group grants him access to the Acme record.

Inherited Grants
Grants that occur when a user, personal or public group, queue, role, or territory inherits access through a role or territory hierarchy,
or is a member of a group that inherits access through a group hierarchy.

Implicit Grants
Grants that occur when non-configurable record-sharing behaviours built into Salesforce Sales, Service, and Portal applications grant access to certain parent and child records. For example, with this default logic, sometimes referred to as built-in sharing, users can view a parent account record

if they have access to its child opportunity, case, or contact record. If those users have access to a parent account record, they can also access its child opportunity, case, and contact records.

Database Architecture

Salesforce stores access grants in three types of tables.

Object Record Tables

Tables that store the records of a specific object, and indicate which user, group, or queue owns each record.

Object Sharing Tables
Tables that store the data that supports explicit and implicit grants. Most objects in your organisation (to see them, from Setup, enter Sharing Settings in the Quick Find box, then select Sharing Settings) get their own Object Sharing table, unless any of
the following conditions are also true:

• The object is a detail in a master-detail relationship. In master-detail relationships, the Object Sharing table for the master object
controls access to the detail object.
• Both organisation-wide default settings (internal and external) are Public Read/Write.
• The object is of a type that doesn't support Object Sharing tables, such as Activities or Files. These objects have their own access
control mechanism.

Group Maintenance Tables
Tables that store the data supporting group membership and inherited access grants. For example, if the Object Sharing table grants Bob explicit access to the Acme account record, Salesforce checks the Group Maintenance tables to see which users inherit record access from Bob and grants users access to the Acme record. These grants are established in advance when you create or change the group (or role, or territory) membership information.
While Object 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 searching, querying, or pulling up a report or list view.

When a user tries to retrieve one or more records, Salesforce generates a SQL statement that searches the Object Record table for records matching the user's search string. If the record exists, Salesforce appends SQL to the statement that joins the Object Records table with the Object Sharing table, and the Object Sharing table with the Group Maintenance tables. Salesforce queries the joined tables for access grants that give the querying user access to the record.