

[Type here]

By Smriti Sharan ([sfdcamlified](#))

Complete Guide of 160+ Scenario Based Salesforce Security

Interview Questions.

Hello, my Name is Smriti Sharan. I am [avid blogger](#) and [youtuber](#). Follow my Blog and youtube to learn various aspect of Salesforce

1. Imagine a financial institution where two departments—loans and investments—operate separately. To maintain confidentiality, employees in the loans department (User A) should not see customer investment records, and vice versa (User B)?

1. Organization-Wide Defaults (OWD): Set the **OWD for the relevant object to "Private."** This setting ensures that records are not shared unless explicitly granted.

2. Role Hierarchy: **Ensure that Users A and B are on the same level** in the role hierarchy or in different branches without a common parent role. This prevents automatic sharing through the role hierarchy.

3. Profile Permissions: **Users A and B should not have "Modify All" or "View All" permissions** for the object, as these permissions override sharing settings and grant access to all records.

4. Sharing Rules: Verify that there are no sharing rules or manual sharing entries that grant access to the records between these two users.

2. In a sales organization, a User A can see the sales records of all salespeople in their region, but User B cannot see records of other regions. What is the reason?

This scenario suggests that User A has higher permissions or access levels compared to User B. The potential reasons include:

[Type here]

By Smriti Sharan ([sfdcamlified](#))

1. Role Hierarchy: User A might be higher in the role hierarchy, allowing them to view records owned by users below them.

2. Profile or Permission Sets: User A could have a profile or permission set with additional permissions, such as "View All" or specific sharing rules.

3. Public Group or Sharing Rule: User A might be part of a public group that has additional sharing rules applied, extending access to certain records.

4. Manual Sharing or Apex Sharing: Records might be shared with User A through manual sharing or programmatically via Apex sharing.

3. A record is shared with User A with read-only access, but User A's profile has edit permissions. Can User A edit the record?

No, User A cannot edit the record. In Salesforce, the intersection of object-level security and record-level security determines access. In this case:

Object-Level Security (Profile Permissions): User A's profile allows editing access of this object.

Record-Level Security (Sharing Settings): The record is shared with read-only access.

Salesforce Rule: When there's a conflict, Salesforce applies the most restrictive setting. Here, the read-only sharing setting restricts User A from editing, despite their profile permissions.

4. User A can edit a record shared with them with read-only access. How is this possible?

"Modify All Data" Permission: A system-wide permission that allows editing any record.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

5. User A has create permission on an object and a record is shared with them with view-only access. Despite not owning the record, they can edit it. Why?

Even if a record is shared with view-only access, User A might still be able to edit it under certain conditions:

1. "Modify All" or "View All" Permissions: If User A has "Modify All" permissions on the object, they can edit all records, including those shared with view-only access.

2. Public Group or Manual Sharing: User A might be part of a public group or have been granted additional permissions through manual sharing.

6. If there is a conflict between object-level permissions and record-level sharing settings, how does Salesforce resolve it?

Salesforce resolves conflicts by applying the most restrictive permission:

Object-Level Permissions (Profiles, Permission Sets): Define the general actions a user can perform on records of an object, such as create, read, edit, and delete.

Record-Level Permissions (Sharing Rules, Manual Sharing): Fine-tune access to individual records.

Most Restrictive Access Rule: If a user's object-level permissions allow editing but a record is shared with them with view-only access, the user cannot edit that specific record.

7. Two users, User A and User B, are in the same role but should not see each other's data due to privacy concerns. How can you implement this restriction in Salesforce?

Even though users are in the same role, you can **restrict their access by setting the Organization-Wide Default (OWD) to "Private"** for the relevant

[Type here]

By Smriti Sharan ([sfdcamlified](#))

objects. This ensures that users can only see records they own unless additional sharing rules or manual sharing is applied.

8. User C needs access to certain account records but should not edit them. These records are currently private. How can you grant User C view-only access?

To grant User C view-only access, you can use sharing rules to share specific account records with a public group or role that includes User C. The sharing rule should specify "Read Only" access to ensure User C cannot edit the records.

9. User D can view all the data of User E, but User E cannot view User D's data. Both users are in different roles but have the same profile. What could be causing this?

This situation could be caused by User D being higher in the role hierarchy than User E. In Salesforce, users in higher roles can view records owned by users in lower roles within their hierarchy. Alternatively, User D might have been granted additional permissions through a permission set or sharing rules..

10. A record owned by User F is transferred to User G. What happens to the sharing settings applied to that record?

When ownership of a record changes, Salesforce retains the existing sharing settings, but it also considers the permissions of the new owner (User G). If User G has broader permissions, they may inherit more access than the previous owner had. Manual sharing by the previous owner is typically revoked unless re-applied by the new owner.

11. User A has "Modify All" permission for the Opportunity object through their profile but can only view a particular opportunity. What could be causing this discrepancy?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

This discrepancy could occur if there are Apex sharing rules or manual sharing settings that restrict access to that particular record. Another possibility is that User H's profile permissions do not override restrictions set by these more specific sharing rules.

12.What is implicit sharing in Salesforce, and how does it affect data visibility between parent and child records?

Implicit sharing automatically provides access to related records based on a user's permissions. If a user has access to a parent record (like an Account), they implicitly gain access to its child records (such as Contacts), and vice versa. This sharing cannot be modified directly but is controlled through the parent record's sharing settings.

13. How do public groups differ from queues in Salesforce, and how are they used in sharing settings?

Public groups are used to extend sharing settings to multiple users, whereas queues are used for record ownership and management. Public groups can include users, roles, and other public groups, and are primarily used to manage record visibility. Queues, however, are used to manage and distribute records (like cases or leads) among users.

14. If a record owner leaves the organization and the record ownership changes, what happens to manual sharing settings previously set by the original owner?

When record ownership changes, manual sharing settings associated with the original owner are typically removed. The new owner must set up new sharing settings as needed.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

15. If a user does not have permission to create records of a certain record type, can they still view records of that type?

Yes, a user can still view records of a record type even if they do not have permission to create records of that type. **Record type permissions control creation but not visibility**, which is managed by object-level and record-level permissions.

16. Two users, User X and User Y, have the same profile with read and edit access. However, User Y cannot edit certain records. What could be the reasons for this discrepancy?

1. Object-Level Permissions (Profiles):

Profiles define what actions a user can perform on records of an object, such as read, create, edit, and delete. In this case, **both users have profiles that grant read and edit permissions on the object.**

2. Organization-Wide Defaults (OWD):

OWDs are the baseline security settings that determine the default access users have to records they do not own. The OWD for an object can be set to "Private," "Public Read Only," or "Public Read/Write."

Cause : OWD Set to Read Only

If the OWD for the object is set to **"Public Read Only,"** then users can see all records but can only edit records they own. This setting overrides the profile's edit permission for records not owned by the user.

To allow User Y to edit records owned by others, you need to change the OWD setting to "Public Read/Write" or use sharing rules to grant specific edit permissions.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

17. A user reports that they cannot see records owned by their subordinates, even though they are higher in the role hierarchy. What could be the reason?

The most likely cause is that the "Grant Access Using Hierarchies" setting is not enabled for the specific object. **This setting allows users higher in the role hierarchy to automatically gain access to records owned by users below them.** If this setting is disabled, the role hierarchy will not influence record visibility for that object.

18. What happens to manual sharing settings when the ownership of a record is transferred to another user?

When the ownership of a record is transferred, **all manual sharing settings associated with the original owner are removed.** The new owner must set up new sharing settings as needed to provide access to other users.

19. Can a user see a specific field on a record if they have view access to the record but not to the field?

No, if field-level security restricts access to a particular field, the user cannot see that field, **even if they have access to the record.** Field-level security controls visibility at the field level, independent of the record-level or object-level permissions.

20. What is permission set group?

Permission Set Groups is a feature that allows Admins to combine multiple permission sets into a single permission set group for user assignment. With the grouping mechanism, admins can truly apply role-based access control for managing user entitlements in Salesforce orgs.

21. Is it possible to bypass Grant Login access using Hierarchies in case of standard objects?

No

[Type here]

By Smriti Sharan ([sfdcamlified](#))

22. Can custom object on detail side has sharing rule?

Custom objects on the detail side of a master-detail relationship cannot have sharing rules, manual sharing, or queues, as these require the Owner field.

23. Why I am not able to find list of Person Account fields in Field Level Security (FLS) settings when navigated to fields on Account Object?

Field Level Security (FLS) of Person Account fields are controlled by Contact Fields. So, if you want to setup FLS of Person Account Fields navigate to fields of Contact and it will be reflected on Person Account.

24. What are types of sharing rules?

Sharing rules are used to **open up the access** to the Salesforce Record

Sharing rule works if OWD of record is either private or public read only

When sharing rule is executed then **behind the scene salesforce created record for Share object**

We cannot share the records with user directly. **We need to add users to public group to share records.**

There are three type of sharing rules:

- Owner based sharing rule
- Criteria based sharing rule. You can't use Apex to create criteria-based sharing rules. Also, criteria-based sharing cannot be tested using Apex.
- Guest user access, based on criteria

Can share record with –

- Public Group
- Roles
- Queues
- Roles and Internal Subordinates

[Type here]

By Smriti Sharan ([sfdcamlified](#))

- Roles , Portal and Internal Subordinates

25. Explain Apex Sharing?

To access sharing programmatically, you must use the share object associated with the standard or custom object for which you want to share. For example, AccountShare is the sharing object for the Account object, ContactShare is the sharing object for the Contact object. In addition, all custom object sharing objects are named as follows, where MyCustomObject is the name of the custom object:

MyCustomObject__Share

A share object includes records supporting all three types of sharing:

- managed sharing
- user managed sharing
- Apex managed sharing

26. What are some considerations for Apex Managed Sharing?

Only users with “Modify All Data” permission can add or change Apex managed sharing on a record

27. What are considerations of sharing object?

Objects on the detail side of a master-detail relationship do not have an associated sharing object.

You cannot create “__share” object by yourself. System create it for us. If sharing setting of an object is “Public Read/Write” system will not create “__share” object, as there is no scope of sharing, all record is open to everybody in org. However, if sharing setting of object is either “Public Read Only” or “Private“, system itself create a “__share” object for us.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

28. Case object has OWD set to private. Now regardless of hierarchies, like top to down (e.g. manager can view lead cases), down to top (e.g. lead can view manager cases), and horizontal (e.g. lead can view lead cases), and cross functionally, all cases should be visible to anyone without change in OWD. How is this possible?

Create a criteria based sharing rule where give access to “Roles and subordinates” to the head of department, this will let everyone access case regardless of hierarchy.

29.Is it possible to create sharing rules for detail object?

No, we can create sharing rules for details objects because they don't have owner field.

30. What are properties of sharing object?

objectNameAccessLevel: The level of access that the specified user or group has been granted for a share sObject

Valid values are:

- Edit
- Read
- All

ParentID: The ID of the object. This field cannot be updated.

RowCause: The reason why the user or group is being granted access. The reason determines the type of sharing, which controls who can alter the sharing record. This field cannot be updated.

UserOrGroupId: The user or group IDs to which you are granting access. A group can be:

- A public group or a sharing group associated with a role.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

- A territory group.

31.Apex run in which mode?

Apex generally runs in system context; that is, the current user's permissions and field-level security are not taken into account during code execution.

32.What effect with sharing makes on user's permission and FLS?

Enforcing sharing rules by using the with sharing keyword **doesn't enforce the user's permissions and field-level security**. Apex code always has access to all fields and objects in an organization, ensuring that code won't fail to run because of hidden fields or objects for a user

33. How to apply object-level and FLS in apex?

Apex doesn't enforce object-level and field-level permissions by default, you can enforce these permissions in your SOQL queries by using WITH SECURITY_ENFORCED.

You can also enforce object-level and field-level permissions in your code by explicitly calling the sObject describe result methods (of [Schema.DescribeSObjectResult](#)) and the field describe result methods (of [Schema.DescribeFieldResult](#)) that check the current user's access permission levels. In this way, you can verify if the current user has the necessary permissions, and only if he or she has sufficient permissions, you can then perform a specific DML operation or a query.

For example, you can call the isAccessible, isCreateable, or isUpdateable methods of Schema.DescribeSObjectResult to verify whether the current user has read, create, or update access to an sObject, respectively.

To check the field-level update permission of the contact's email field before updating it:

1. if (Schema.sObjectType.Contact.fields.Email.isUpdateable()) {
2. // Update contact phone number

[Type here]

By Smriti Sharan ([sfdcamplified](#))

3. }

To check the field-level create permission of the contact's email field before creating a new contact:

```
1. if (Schema.sObjectType.Contact.fields.Email.isCreateable()) {  
2. // Create new contact  
3. }
```

To check the field-level read permission of the contact's email field before querying for this field:

```
1. if (Schema.sObjectType.Contact.fields.Email.isAccessible()) {  
2. Contact c = [SELECT Email FROM Contact WHERE Id= :Id];  
3. }
```

To check the object-level permission for the contact before deleting the contact:

```
1. if (Schema.sObjectType.Contact.isDeletable()) {  
2. // Delete contact  
3. }
```

34.What is the use of striplnaccessible ?

Use the `striplnaccessible` method to enforce field- and object-level data protection. This method can be used to strip the fields and relationship fields from query and subquery results that the user can't access. The method can also be used to remove inaccessible sObject fields before DML operations to avoid exceptions and to sanitize sObjects that have been deserialized from an untrusted source.

35.Does striplnaccessible support AggregateResult object?

The `striplnaccessible` method doesn't support `AggregateResult` SObject. If the source records are of `AggregateResult` SObject type, an exception is thrown.

36. What is User Managed Sharing ?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

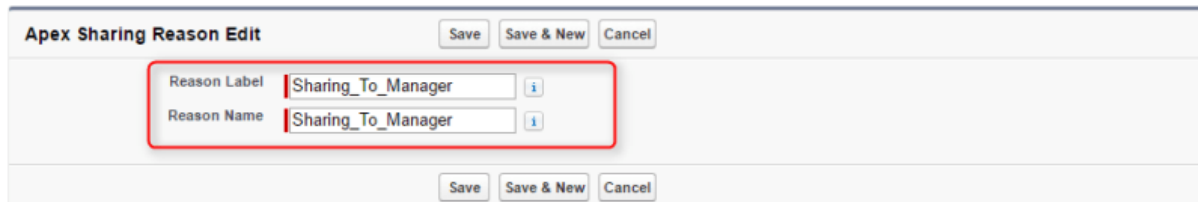
In User Managed Sharing is, a user who has full access on the record shares the records with others, but when the record owner is changed, this record will be removed from Sharing table. Similarly, when apex sharing is defined as “Manual” on RowCause, it will remove record from sharing table when the record owner is changed.

To resolve this issue, we need to define Apex Sharing Reason on Rowcause while writing Apex Sharing.

Apex Sharing Reason

New Apex Sharing Reason

Apex sharing reasons are used by developers when adding sharing to a record programmatically. Using an apex sharing reason prevents standard users from de the developer to track why they added the sharing.



Example Code:

```
trigger DynamicContent_AT on Dynamic_Content__c(after insert) {  
    // We only execute the trigger after a Dynamic_Content__c record has been inserted  
    // because we need the Id of the Dynamic_Content__c record to already exist.  
    if(trigger.isInsert){  
        // Dynamic_Content__Share is the "Share" table that was created when the  
        // Organization Wide Default sharing setting was set to "Private".  
        // Holding list of Dynamic_Content__Share records.  
        List<Dynamic_Content__Share> dcSharesList = new List<Dynamic_Content__Share>();  
  
        // Sharing Dynamic Content records to Chapter Staff.  
        for(Dynamic_Content__c currRec : trigger.new){  
            // Create a new Dynamic_Content__Share record to be inserted in to the Dynamic_Content__Share table.  
            Dynamic_Content__Share dcShare = new Dynamic_Content__Share();  
  
            // Populate the Dynamic_Content__Share record with the ID of the record to be shared.  
            dcShare.ParentId = currRec.Id;  
  
            // Specify the Chapter_Staff user id for sharing records.  
            dcShare.UserOrGroupId = currRec.Chapter_Staff__c;  
  
            // Granting Edit Permission for the Chapter Staff.  
            dcShare.AccessLevel = 'edit';  
  
            // Specify the mode of sharing.  
            // (Shairng To Manager __c is the Apex Sharing Reason that we defined earlier.)  
            dcShare.RowCause = Schema.Dynamic_Content__Share.RowCause.Shairng_To_Manager__c;  
  
            // Add the new Share record to the list of new Share records.  
            dcSharesList.add(dcShare);  
        }  
  
        // Insert all of the newly created Share records and capture save result  
        Database.SaveResult[] dcShareInsertResult = Database.insert(dcSharesList,false);  
    }  
}
```

[Type here]

By Smriti Sharan ([sfdcamplified](#))

Since we have defined Apex Sharing Reason on Custom Object sharing, it will keep Share table records updated whenever record owner is changed. So, still granted user can access the records without any issue.

37.Explain Apex Sharing for Standard Objects?

Standard objects don't support Apex Sharing Reason. So, while sharing standard object records, by default you must define RowCause is "Manual".

```
trigger Account_AT on Account(after insert,After Update) {
    // We only execute the trigger after a Account record has been inserted
    // because we need the Id of the AccountShare record to already exist.
    if(trigger.isInsert){
        // AccountShare is the "Share" table that was created when the
        // Organization Wide Default sharing setting was set to "Private".
        // Holding list of AccountShare records.
        List<AccountShare> accountList = new List<AccountShare>();

        // Sharing records to Chapter Staff.
        for(Account currRec : trigger.new)
        {
            if(trigger.isInsert || (Trigger.isUpdate && currRec.OwnerId != Trigger.oldMap.get(currRec.id).OwnerId))
            {
                // Create a new AccountShare record to be inserted in to the AccountShare table.
                AccountShare accShare = new AccountShare();

                // Populate the AccountShare record with the ID of the record to be shared.
                accShare.ParentId = currRec.Id;

                // Specify the Chapter_Staff user id for sharing records.
                accShare.UserOrGroupId = currRec.Chapter_Staff__c;

                // Granting Edit Permission for the Chapter Staff.
                accShare.AccessLevel = 'edit';

                // Specify the mode of sharing.
                accShare.RowCause = Schema.AccountShare.RowCause.Manual;

                // Add the new Share record to the list of new Share records.
                accountList.add(accShare);
            }
        }

        // Insert all of the newly created Share records and capture save result
        Database.SaveResult[] accShareInsertResult = Database.insert(accountList,false);
    }
}
```

38. Case object has OWD set to private. Now regardless of hierarchies, like top to down (e.g. manager can view lead cases), down to top (e.g. lead can view manager cases), and horizontal (e.g. lead can view lead cases), and cross functionally, all cases should be visible to anyone without change in OWD. How is this possible?

Create a criteria based sharing rule where give access to "Roles and subordinates" to the head of department , this will let everyone access case regardless of hierarchy.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

39.What are the considerations while using with sharing?

- If the class is not declared as With Sharing or Without Sharing then the class is by default taken as Without Sharing.
- Both inner classes and outer classes can be declared as With Sharing.
- If inner class is declared as With Sharing and top level class is declared as Without Sharing, then by default entire context will run in With Sharing Context.
- If a class is not declared as With/Without Sharing and if this class is called by another class in which sharing rules is enforced then both the classes run with With Sharing.
- Outer class is declared as With Sharing and inner class is declared as Without Sharing, then inner class runs in Without Sharing Context only (Inner class don't take the Sharing properties from outer class).

40.What is the difference between apex managed sharing and with sharing?

Apex Managed Sharing is used to grant the access to the records. It is about programmatically configuring sharing rules. Keyword "With Sharing" is used to respect the current user sharing rule.

41.What are considerations while using apex managed sharing?

- If record owner changes, then sharing created through apex managed sharing are maintained but if user share record manually, then record sharing will be lost if owner changes.
- User with "modify All Data" can only add, edit or delete records in share table.

42.What are the limitations of manual sharing?

- Manual Sharing cannot be stricter than Organization Wide Defaults.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

- Manual Sharing is only available on individual records, it is not available for all records of a certain object.
- Only applicable on records that have Private or Public Read Only access in OWD.
- When setting Automatic and Manual Sharing users and admins should define if the security should be extended to related records.

43.What is With sharing and without sharing?

With Sharing: It means “with Security Settings enforced”. If you declare a class as a With Sharing, Sharing rules given to the current user will be taken into the consideration. This, pertains to only respecting OWDs and Sharing Rules. We cannot “automatically” enforce field level security or profile permissions with “with sharing,”

Without Sharing: If you declare a class as a Without Sharing, then this Apex class runs in system mode which means Apex code has access to all the objects and field irrespective of current users sharing rules, field level security and Object permissions.

44.Difference between role and profile?

- Profiles help to control object privileges such as CRED (Create, Read, Edit, Delete). They also contain system permissions that a user can carry out such as exporting data.
- Roles help with sharing records across an organization. They work in a hierarchical fashion, giving users access to records that are owned by people lower down in the hierarchy.
- A user can only have a single Profile and Role assigned to them.

45.A can see B data but B cannot see A data. Why?

- Check following conditions
- 1 – Both users have same profile?
- 2. Check Role Hierarchy of A and B.
- 3. Check Sharing rule

[Type here]

By Smriti Sharan ([sfdcemplified](#))

46. User A has the button CLONE visible on Accounts, User B can not see the button CLONE on Accounts. Why?

Check following conditions

- 1 – Both users have same profile?
- 2 – There is no Custom Permission Set assigned to any user
- 3 – Do you have record types for the Account Object and associated page layouts for those Record Types
- 4 – You have checked the page layout and Clone button is added to the Page Layout.

47. We have 2 users A and B under same profile and Role. How can we restrict records of A to B and Vice versa?

In profile set **View all data and modify all data permission to 'false'**. This will **restrict user to access data created by other users**.

48. There are 100 users. 90 users can read records, 10 users can update records?

Create two profiles:

First profile give read permission and add 90 users.

Second profile give read and edit permission and add 10 users.

49. What are limitations of OWD model?

OWD narrows down the access. Cannot open access

50. There are five users under one profile and only one user sees all the data. Account is private. Why?

Create permission set with view all access to accounts and assign permission set to specific user

[Type here]

By Smriti Sharan ([sfdcamlified](#))

51. What is SAML?

Security Assertion Markup Language (SAML) is an open standard that enables single sign-on (SSO). By making a range of resources accessible with just one set of login credentials, you can provide seamless access to resources and eliminate insecure password proliferation.

52.What is implicit sharing?

Implicit sharing is automatic. You can neither turn it off, nor turn it on — it is native to the platform.

Parent implicit sharing provide read-only access to parent records (Account only), when user has access to children record, such as: Opportunities, Cases, or Contacts for that account. This does not mean the user must be the record owned of the child record.

When user have access to a record from other objects (NOT opportunity, case, or contact) that have lookup to Account, user will see the Account Name only, but not to access Account detail – this include Account lookup to the Parent Account, child account owner will see Parent Account Name only.

Child Implicit Sharing is ability for Account owner to access child records (contacts, opportunities, and cases), even they are not owned by Account owner. The account owner's role determines the level of access to child records (read-only or read/write).

53.What is System.RunAs?

Generally, all Apex code runs in system mode, where the permissions and record sharing of the current user are not taken into account. The system method runAs enables us to write test methods that change the user context to an existing user or a new user so that the user's record sharing is enforced.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

The runAs method doesn't enforce user permissions or field-level permissions, only record sharing. We can use runAs only in test methods.

54. If user has view access on report folder but in profile he does not have access to dashboard then will user be able to access the dashboard?

A. No, On the profile level user should have View Dashboards in Public Folders— "View and access dashboards in public folders, which does not include others' personal folders" apart from view access at report level for him to view the dashboard.

55. What is a share object?

Every object has a share object. **Share object is a junction between your object and the user you want to share the record**. To access sharing programmatically, you must use the **share object associated with the standard or custom object**

For example, AccountShare is the sharing object for the Account object

Note: **Objects on the detail side of a master-detail relationship do not have an associated sharing object.**

56. In the Salesforce, you are required to ensure that users can only log in from a specific set of IP addresses. How would you achieve this at the organization level?

To restrict users from logging in only from a specific set of IP addresses, you would use the "Login IP Ranges" feature at the profile or permission set level in Salesforce.

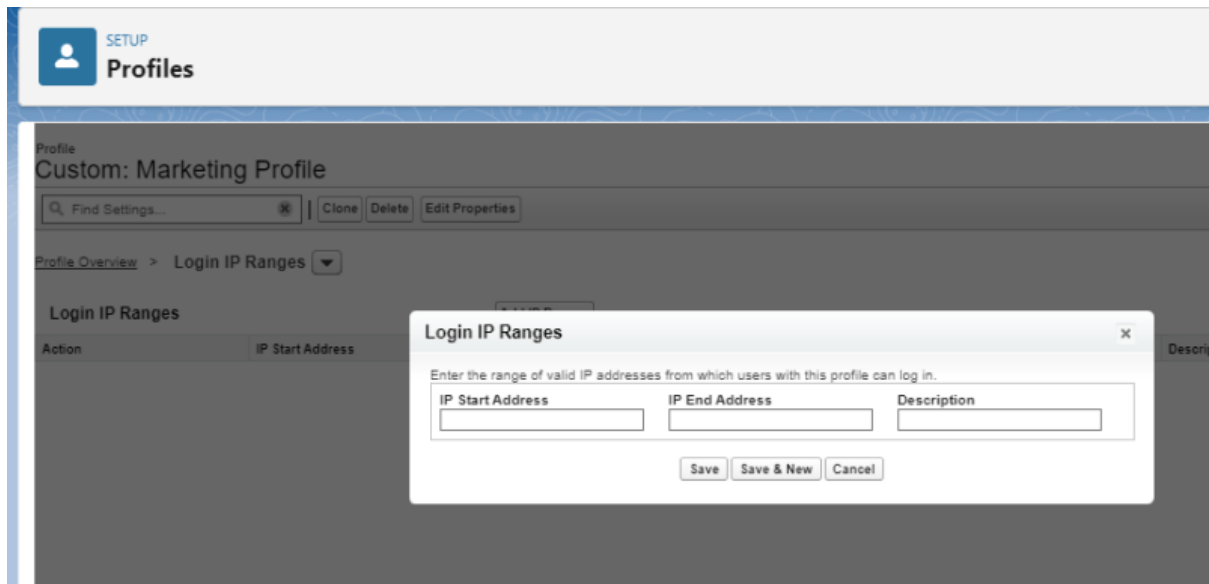
To set this: they

1. Navigate to the profile (or permission set) you want to configure.
2. Under the "Login IP Ranges" section, you can specify the start and end of the allowed IP range.
3. Save the settings.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

It's crucial to note that while "Login IP Ranges" restricts where a user can log in from, it doesn't restrict when they can log in. For that, you'd use "Login Hours".



57. Your organization wants to enforce a policy where users are required to change their passwords every 30 days and the password must be at least 10 characters long. How would you implement this in Salesforce at the organization level?

In Salesforce, under "Password Policies", you have the ability to define the lifespan of a user's password and the complexity of the password.

Here's how you'd set it up:

From the Salesforce setup area, navigate to Security Controls > Password Policies.

Set 'Password Expires In' to 30 days.

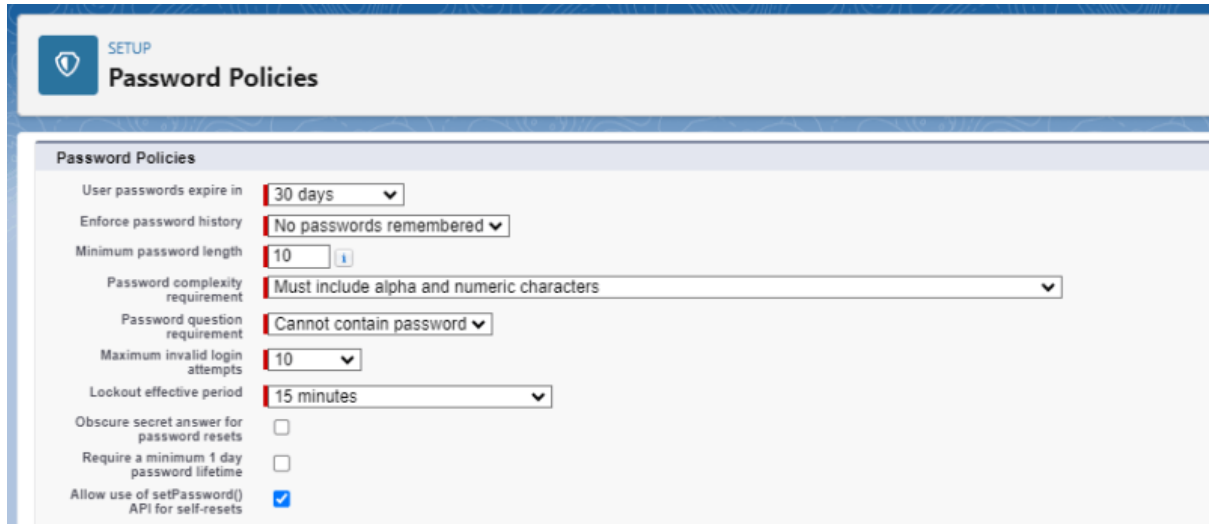
Set 'Minimum Password Length' to 10 characters.

Save the settings.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

With these settings in place, users would be prompted to change their password every 30 days, and the system would enforce the requirement of a password being at least 10 characters long.

A screenshot of the Salesforce 'Password Policies' setup page. The page has a blue header with a shield icon and the word 'SETUP'. Below the header, the title 'Password Policies' is displayed. The main content area is titled 'Password Policies' and contains several configuration options: 'User passwords expire in' set to '30 days', 'Enforce password history' set to 'No passwords remembered', 'Minimum password length' set to '10', 'Password complexity requirement' set to 'Must include alpha and numeric characters', 'Password question requirement' set to 'Cannot contain password', 'Maximum invalid login attempts' set to '10', and 'Lockout effective period' set to '15 minutes'. There are also three unchecked checkboxes: 'Obscure secret answer for password resets', 'Require a minimum 1 day password lifetime', and 'Allow use of setPassword() API for self-resets' which is checked.

58. In Salesforce, how would you restrict users from accessing the Salesforce application outside of regular business hours, say from 9 AM to 5 PM?

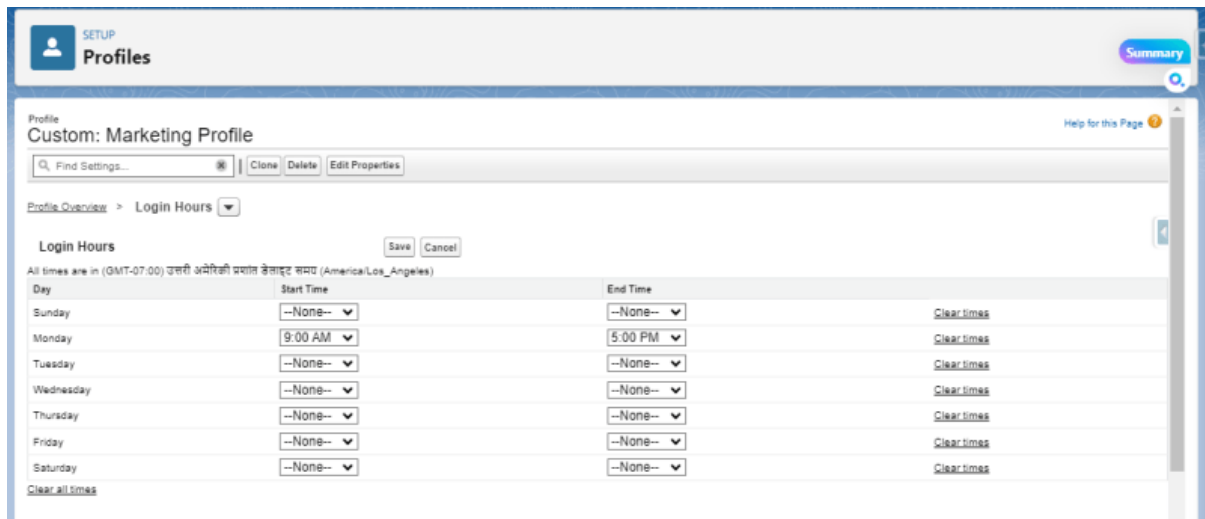
In Salesforce, the “Login Hours” feature allows you to specify the time frames during which users can log into the system based on their assigned profile.

Here’s how you’d set it:

1. Navigate to Setup
2. Search for and select the desired profile.
3. In the profile’s settings, scroll to “Login Hours.”
4. Here, you can set the permitted login start and end times for each day of the week.
5. For your scenario, you’d set the start time to 9 AM and the end time to 5 PM for weekdays.

[Type here]

By Smriti Sharan ([sfdcamlified](#))



59. have sensitive data in certain Salesforce objects, and you want to monitor and be notified if there are unexpected or large data exports. How would you set up this monitoring at the organization level in Salesforce?

To monitor and be notified of unexpected or large data exports in Salesforce, you would use the “Event Monitoring” tool, particularly the “Report Export” event type.

Here’s how to set it up:

1. Go to Setup in Salesforce.
2. In the Quick Find box, type “Event Monitoring.”
3. Select the “Event Manager.”
4. From here, you can view different event logs, including the “Report Export” log.
5. You can set up alerts to notify you when certain thresholds are exceeded, such as when a user exports a large volume of data.

By monitoring the “Report Export” event, you’ll get insights into who is exporting data, what data they’re exporting, and when they’re doing it. This can be crucial in ensuring sensitive data isn’t being accessed or exported inappropriately.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

Name	Subscription Channel	Type	Description	Streaming
API Anomaly Event	/event/ApiAnomalyEvent	Event Monitoring	Track anomalies in how users make API calls	
API Event	/event/ApiEventStream	Event Monitoring	Track user API queries in your org	
Bulk API Result Event	/event/BulkApiResultEvent	Event Monitoring	Track when a user downloads the results of a Bulk API request	
Concurrent Long Running Apex Err...	/event/ConcurLongRunApexEr...	Event Monitoring	Track when a Concurrent Long Running Apex error has occurred	
Credential Stuffing Event	/event/CredentialStuffingEvent	Event Monitoring	Track when a user successfully logs in to Salesforce during an identified creden...	
File Event	/event/FileEvent	Event Monitoring	Track file activity. For example, track when a user downloads or previews a file	
Identity Provider Event	Not applicable	Event Monitoring	Track identity provider activities	Not applici
Identity Verification Event	Not applicable	Event Monitoring	Track when users verify their identity	Not applici
Lightning URI Event	/event/LightningUriEventStream	Event Monitoring	Track when a user creates, accesses, updates, or deletes a record in Salesforce ...	
List View Event	/event/ListViewEventStream	Event Monitoring	Track when a user accesses data with list views	
Login Event	/event/LoginEventStream	Event Monitoring	Track when a user logs in to your org	
LoginAs Event	/event/LoginAsEventStream	Event Monitoring	Track when an admin logs in to your org as another user	
Logout Event	/event/LogoutEventStream	Event Monitoring	Track when a user clicks Log Out in the Salesforce UI	
Permission Set Event (Beta)	/event/PermissionSetEvent	Event Monitoring	Track when users are assigned the Modify All Data or View All Data permission ...	
Report Anomaly Event	/event/ReportAnomalyEvent	Event Monitoring	Track anomalies in how users run or export reports	
Report Event	/event/ReportEventStream	Event Monitoring	Track when a user accesses or exports data with reports	
Session Hijacking Event	/event/SessionHijackingEvent	Event Monitoring	Track when an unauthorized user gains ownership of a Salesforce user's sessio...	
URI Event	/event/UriEventStream	Event Monitoring	Track when a user creates, accesses, updates, or deletes a record in Salesforce ...	

Setup > Event Manager				
Report Event				
API Name	Subscription Channel	Type	Description	Streaming Data
ReportEvent	/event/ReportEventStream	Event Monitoring	Track when a user accesses or exports data with reports	Storing Data
				✓

60. In your Salesforce organization, you've observed that certain sensitive fields are being accessed by more users than expected. How would you set up a mechanism to track which users are viewing these sensitive fields?

The feature you'd use in Salesforce to track access to specific fields is called "Field Audit Trail".

With Salesforce's Field Audit Trail, you can get a detailed history of who has viewed or changed specific fields in records. It provides an added layer of visibility, especially for sensitive data.

Here's how to set it up:

1. Navigate to Setup.
2. Search for the object that contains the sensitive field you want to monitor
3. Under the object, go to "Fields & Relationships."

[Type here]

By Smriti Sharan ([sfdcamlified](#))

4. Find the specific field you want to track and click on it.

5. Click on “Set History Tracking.”

6. Check the box next to “Enable” for that field.

Once this is done, every time the field is viewed or edited, Salesforce will create a history record indicating which user accessed the field and when.

☒ Enable Account History

This page allows you to select the fields you want to track on the Account History related list. Whenever a user modifies any of the fields selected below, the old and new field values are added time, nature of the change, and user making the change. Note that multi-select picklist and large text field values are tracked as edited; their old and new field values are not recorded.

Save Cancel

[Deselect all fields](#)

Track old and new values

Account	<input type="checkbox"/>	Account Currency	<input type="checkbox"/>
Account Name	<input checked="" type="checkbox"/>	Account Number	<input type="checkbox"/>
Account Owner	<input type="checkbox"/>	Account Record Type	<input type="checkbox"/>
Account Site	<input type="checkbox"/>	Account Source	<input type="checkbox"/>
Active	<input type="checkbox"/>	Annual Revenue	<input type="checkbox"/>
Balance	<input type="checkbox"/>	Billing Address	<input type="checkbox"/>
Clean Status	<input type="checkbox"/>	Co-Termination Event	<input type="checkbox"/>
Combine Asset Quantities	<input type="checkbox"/>	Combine Co-Termed Contracts	<input type="checkbox"/>
Contract Co-Termination	<input type="checkbox"/>	Country	<input type="checkbox"/>
Customer ID	<input type="checkbox"/>	Customer Priority	<input type="checkbox"/>

61.You’ve been tasked to keep track of changes made to the metadata and setup configurations in your Salesforce organization. How would you monitor these changes, especially for critical configurations, using Salesforce’s built-in features?

The Setup Audit Trail in Salesforce provides a mechanism to track changes made in the setup area of your Salesforce organization. It helps you understand who did what and when, which is vital for maintaining security and for understanding the cause of any unintended setup changes.

Here’s how you can view the Setup Audit Trail:

1. Navigate to Setup

2. In the Quick Find box, type “Setup Audit Trail.”

[Type here]

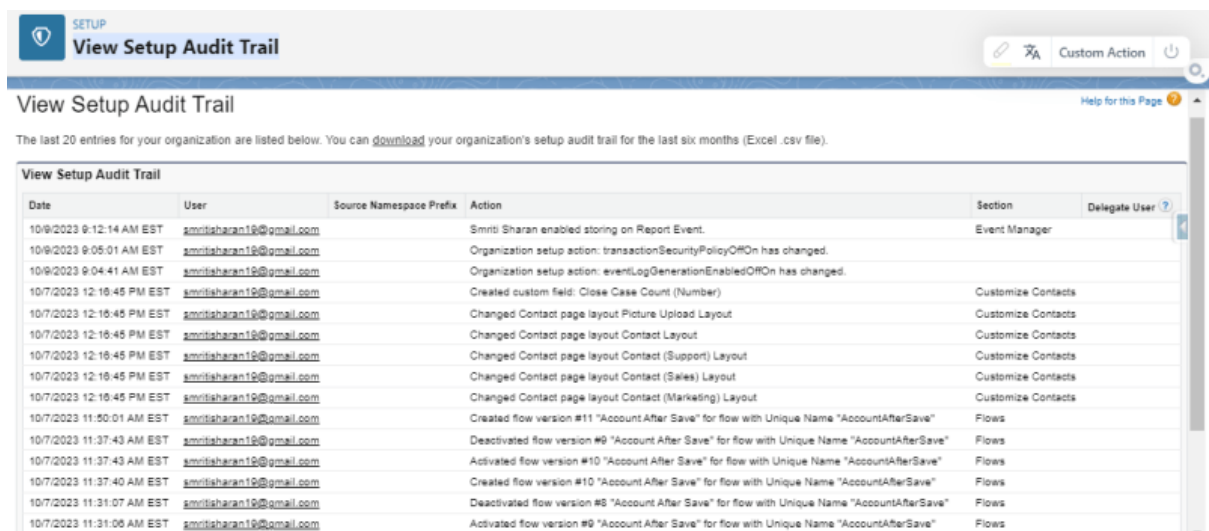
By Smriti Sharan ([sfdcamlified](#))

3. Click on the “View Setup Audit Trail” link.

Here you’ll find the last 20 changes made in your setup. You’ll see information like who made the change, what the change was, and when it was made.

Additionally, Salesforce also offers the ability to download the last six months of setup audit trail changes as a CSV file.

Using the Setup Audit Trail, you can monitor crucial configuration changes, such as changes to security settings, field customizations, and user permissions.



Date	User	Source Namespace Prefix	Action	Section	Delegate User
10/9/2023 9:12:14 AM EST	smritisharan19@gmail.com		Smriti Sharan enabled storing on Report Event.	Event Manager	
10/9/2023 9:05:01 AM EST	smritisharan19@gmail.com		Organization setup action: transactionSecurityPolicyOn has changed.		
10/9/2023 9:04:41 AM EST	smritisharan19@gmail.com		Organization setup action: eventLogGenerationEnabledOn has changed.		
10/7/2023 12:16:45 PM EST	smritisharan19@gmail.com		Created custom field: Close Case Count (Number)	Customize Contacts	
10/7/2023 12:16:45 PM EST	smritisharan19@gmail.com		Changed Contact page layout Picture Upload Layout	Customize Contacts	
10/7/2023 12:16:45 PM EST	smritisharan19@gmail.com		Changed Contact page layout Contact Layout	Customize Contacts	
10/7/2023 12:16:45 PM EST	smritisharan19@gmail.com		Changed Contact page layout Contact (Support) Layout	Customize Contacts	
10/7/2023 12:16:45 PM EST	smritisharan19@gmail.com		Changed Contact page layout Contact (Sales) Layout	Customize Contacts	
10/7/2023 12:16:45 PM EST	smritisharan19@gmail.com		Changed Contact page layout Contact (Marketing) Layout	Customize Contacts	
10/7/2023 11:50:01 AM EST	smritisharan19@gmail.com		Created flow version #11 "Account After Save" for flow with Unique Name "AccountAfterSave"	Flows	
10/7/2023 11:37:43 AM EST	smritisharan19@gmail.com		Deactivated flow version #9 "Account After Save" for flow with Unique Name "AccountAfterSave"	Flows	
10/7/2023 11:37:43 AM EST	smritisharan19@gmail.com		Activated flow version #10 "Account After Save" for flow with Unique Name "AccountAfterSave"	Flows	
10/7/2023 11:37:40 AM EST	smritisharan19@gmail.com		Created flow version #10 "Account After Save" for flow with Unique Name "AccountAfterSave"	Flows	
10/7/2023 11:31:07 AM EST	smritisharan19@gmail.com		Deactivated flow version #8 "Account After Save" for flow with Unique Name "AccountAfterSave"	Flows	
10/7/2023 11:31:06 AM EST	smritisharan19@gmail.com		Activated flow version #9 "Account After Save" for flow with Unique Name "AccountAfterSave"	Flows	

62. Your company has just undergone an IT audit, and for compliance reasons, they’ve requested that all users in your Salesforce organization have their passwords expired, forcing everyone to set a new password on their next login. How would you achieve this in Salesforce?

To expire all passwords and force every user to reset their password upon the next login, you can use the “Expire All Passwords” feature in Salesforce.

1. Navigate to Setup.

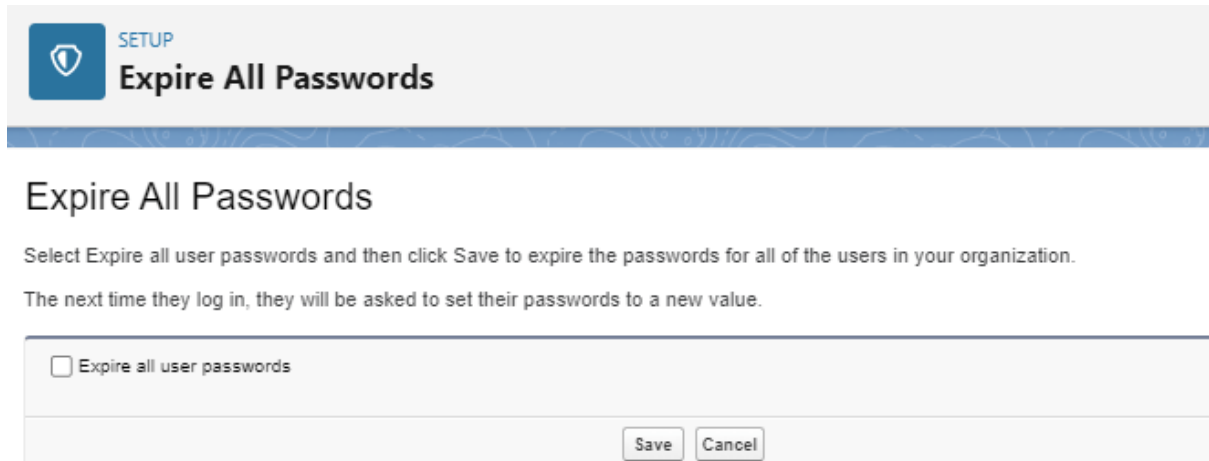
2. In the Quick Find box, type “Expire All Passwords”.

[Type here]

By Smriti Sharan ([sfdcamplified](#))

3. There's an option to "Expire All Passwords". Clicking on this button will expire passwords for all users.

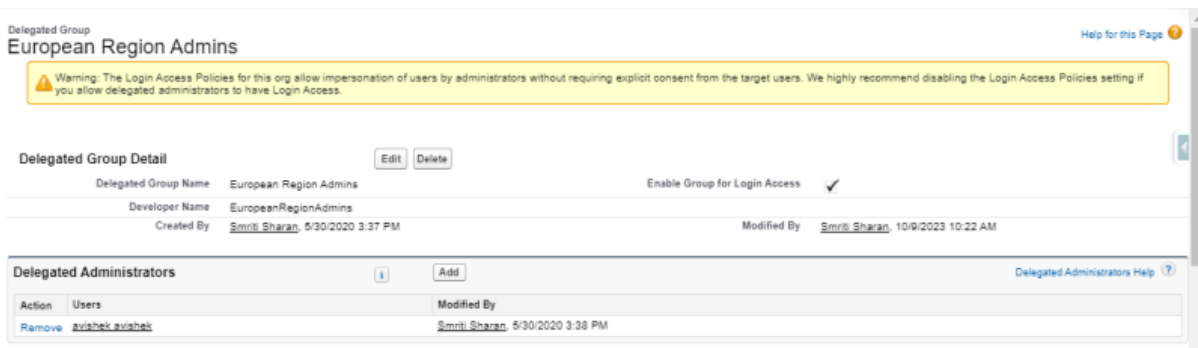
4. Users will be prompted to set a new password the next time they attempt to log in.



The screenshot shows the 'SETUP Expire All Passwords' interface. It includes a title bar with a shield icon and the text 'SETUP Expire All Passwords'. Below the title bar, there is a heading 'Expire All Passwords' and a paragraph of instructions: 'Select Expire all user passwords and then click Save to expire the passwords for all of the users in your organization. The next time they log in, they will be asked to set their passwords to a new value.' A checkbox labeled 'Expire all user passwords' is present, and at the bottom right, there are 'Save' and 'Cancel' buttons.

63. Your company is expanding and has recently opened a new regional office in Europe. As an admin, you don't want to be responsible for user management for this new office. How can you empower the European regional manager to manage users?

By setting up a Delegated Administrator You can specify that the regional manager can manage users with a specific role or within a particular role hierarchy, like all roles under "European Regional Sales."

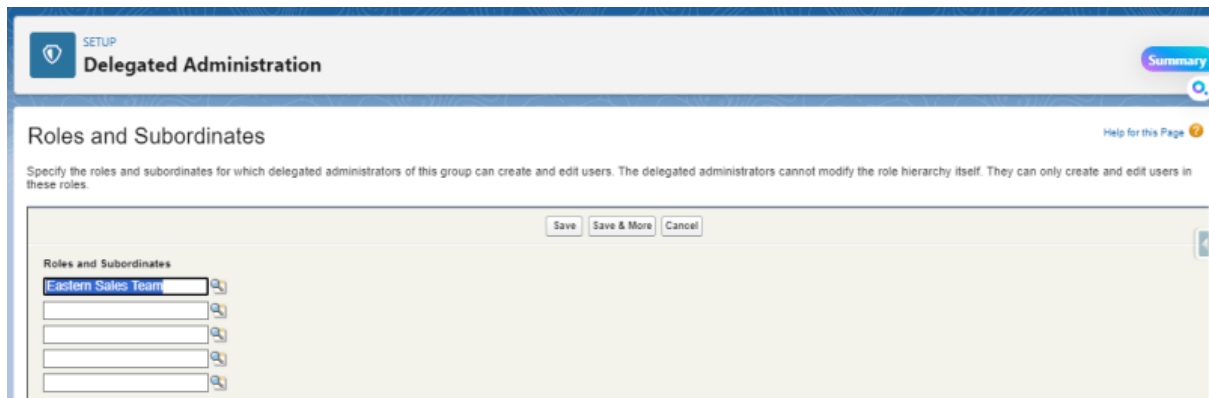


The screenshot displays the 'European Region Admins' Delegated Group configuration page. At the top, it shows the group name 'European Region Admins' and a warning message: 'Warning: The Login Access Policies for this org allow impersonation of users by administrators without requiring explicit consent from the target users. We highly recommend disabling the Login Access Policies setting if you allow delegated administrators to have Login Access.' Below the warning, the 'Delegated Group Detail' section shows the group name, developer name 'EuropeanRegionAdmins', and creation/modification details. The 'Enable Group for Login Access' checkbox is checked. The 'Delegated Administrators' section shows a table with one administrator: 'avishai.avishai'.

Action	Users	Modified By
Remove	avishai.avishai	Smriti Sharan, 5/30/2020 3:38 PM

[Type here]

By Smriti Sharan ([sfdcamlified](#))



64. Your marketing team needs to frequently update picklist values for a custom field named “Campaign Type.” However, you want to ensure they don’t make modifications to other fields. How can you achieve this?

Create a delegated group for the marketing team and grant them permission to modify picklist values. Assign the “Campaign Type” field to this delegated group.

65. You have delegated user management to a team lead. However, the team lead reports that he cannot view certain profiles and roles while assigning them to new users. Why might this be, and how can you resolve it?

Delegated administrators can only assign users to profiles and roles that are specified in their delegated administration group. Ensure that the needed profiles and roles are included in the configuration of the delegated group for the team lead.

66. A delegated admin is reporting that they are unable to edit the login hours for users they are managing. What might be the reason?

Login hours are controlled by profiles. Even if a user is a delegated admin for managing certain users, they can’t change profile settings, including login hours, unless they have the necessary permissions on profiles.

[Type here]

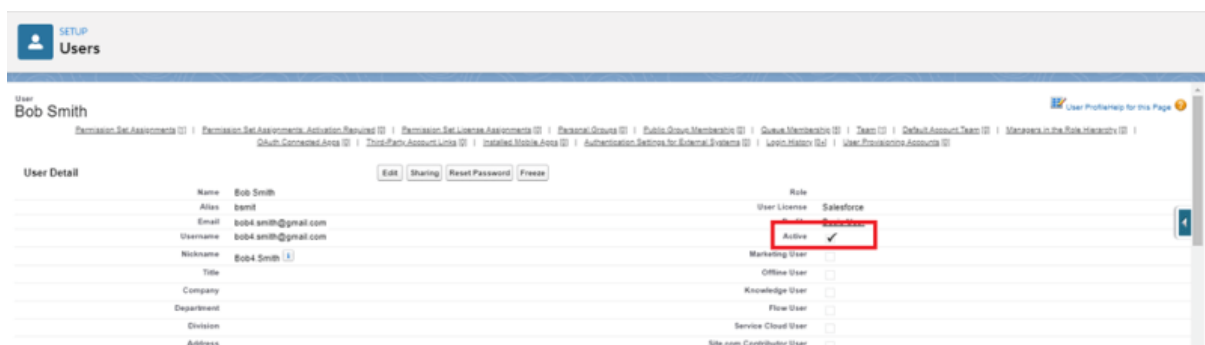
By Smriti Sharan ([sfdcamlified](#))

67.Can a delegated admin modify sharing settings or create sharing rules for the objects they have access to?

No, delegated administration does not provide access to modify sharing settings or create sharing rules. It focuses on administrative tasks like user management and not on security or sharing architecture.

68.How to delete a Salesforce user?

Salesforce user cannot be deleted. They can only be deactivated.

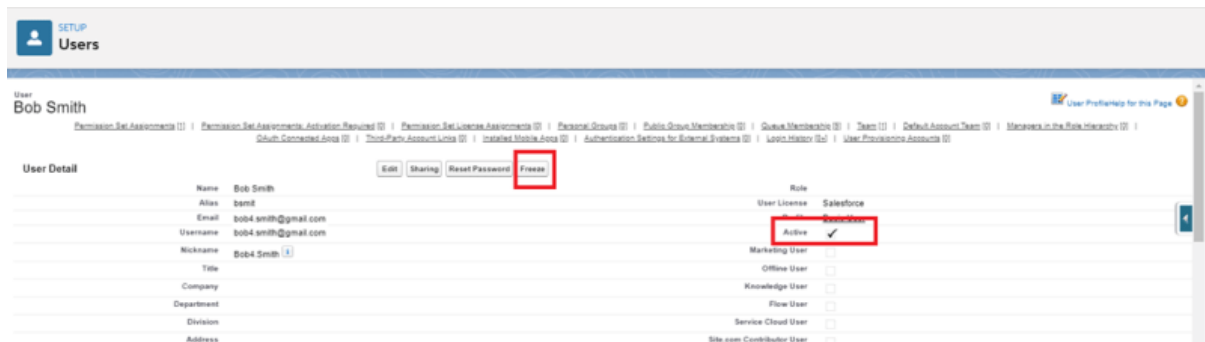


69.What is the difference between freeze user and deactivate a user in Salesforce?

- Freezing user accounts does not make their user licenses available for the user in our org. To make their user license available, deactivate the account.
- Deactivating the user or freezing the user will not allow them to log in to salesforce org.
- In a scenario when we do not want a user to login into Salesforce org for some brief time then instead of deactivating the user we can freeze the user so that they cannot log in into the salesforce for that period.

[Type here]

By Smriti Sharan ([sfdcamlified](#))



70. What is view all and modify all in salesforce?

View all and Modify all permission ignore all the security setting and allow a user to view all the data and modify all the data in the salesforce org irrespective of his role and access.

View all and Modify all access should not be given to any regular user in the salesforce org.

Standard Object Permissions						
	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Accounts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
App Analytics Query Requests	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assets	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

71.I want to delete 10,000 customer records but do not want anyone else to recover them. What can I do?

Salesforce makes it easy to bulk delete records permanently using the **hard delete option**. The difference between delete and hard delete options is that the former sends the deleted records to the Salesforce recycle bin, where it remains for 15 days. The hard delete erases all records permanently from the Salesforce system with no way to recover it.

72.As a Salesforce administrator, how would you give access to users to knowledge articles?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

We can assign user knowledge license by changing the status of the knowledge user checkbox to 'true' on the user detail page. This ensures that the specific user has access to all knowledge articles."

73. Assume a manager who oversees a team of 20 users is leaving the organization. What happens if I inactivate the manager?

The best approach is to deactivate the manager's account and reassign the team to another manager. To prevent users from logging into the account during the reassignment, we freeze the team member's accounts temporarily by using the "freeze" button on the user record. Once the reassignment is over, we can deactivate the freeze button and continue with the new hierarchy.

74.What if user is logged in when their login hours end?

User can continue to their page but they can not do any action. Like they can not create or edit operation.

75.Your organization has a group of users called "Contract Reviewers". They should be able to view and edit all contracts in Salesforce, regardless of who owns them, but they should not be able to delete any contracts. How would you ensure that this group has the correct access to the Contract object?

To give the "Contract Reviewers" the ability to view all contracts, regardless of ownership, but not delete them:

1. Profiles:

- Go to the profile associated with the "Contract Reviewers" group.
- Locate the Contract object permissions.
- Ensure the "Read" and "Edit" permissions are checked.
- Also, ensure that "View All" is checked, so they can view all contracts.
- Ensure "Delete" and "Modify All" are unchecked, so they cannot delete contracts or override other specific permissions.

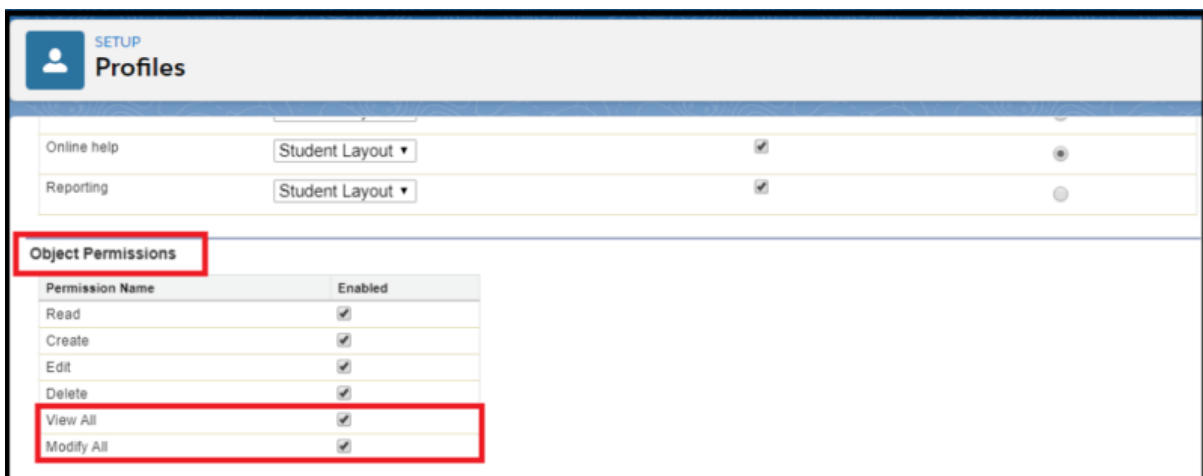
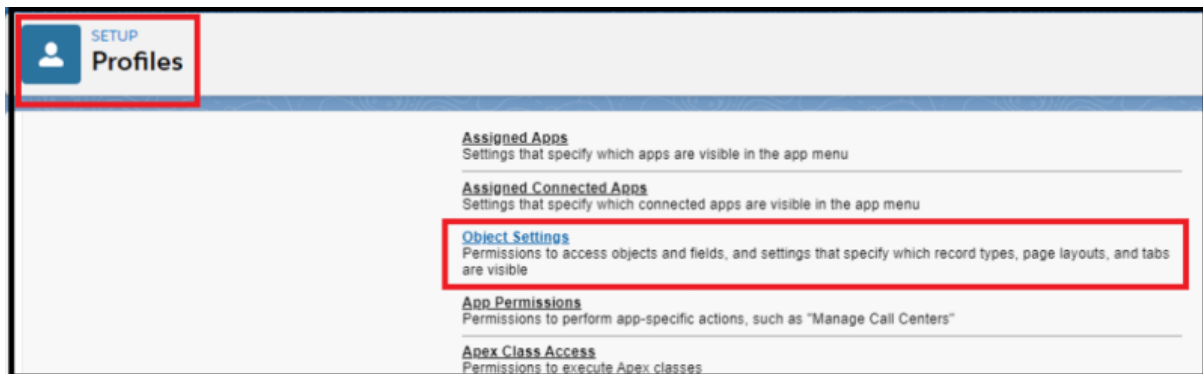
[Type here]

By Smriti Sharan ([sfdcamplified](#))

2. Object Settings:

– Verify in object settings that they have access to the Contract object and its related lists, page layouts, and other associated components.

Using “View All” gives users the ability to view all records of an object, whereas “Modify All” would give them full access, including delete permissions. By providing “View All” but not “Modify All”, you ensure they can view and edit but not delete contracts.



76. You have two teams: Sales and Support. Both teams work with the Account object. The Sales team should see only Accounts where the “Type” is set to “Prospect”, while the Support team should see only Accounts where the “Type” is set to “Customer”. How would you configure this in Salesforce to ensure each team sees only the appropriate Accounts?

1. Profiles:

[Type here]

By Smriti Sharan ([sfdcamlified](#))

- Create two profiles: one for Sales and one for Support.
- Ensure both profiles have at least “Read” access to the Account object.

2. Organization-Wide Defaults (OWD):

- Set the OWD for Accounts to “Private”. This ensures that, by default, users can only see Accounts they own.

3. Criteria-Based Sharing Rules:

- For Sales: Create a criteria-based sharing rule where if the “Type” of Account is “Prospect”, share the record with the Sales profile with Read access.
- For Support: Create another criteria-based sharing rule where if the “Type” of Account is “Customer”, share the record with the Support profile with Read access.

77.A user in your organization should be able to view all the records of the “Lead” object but should not be able to see a specific sensitive field called “Lead Score”. How would you configure this in Salesforce to restrict visibility to the “Lead Score” field for this user?

We can achieve this using Field Level Security

78.Your organization wants to ensure that while all sales reps can view the ‘Discount’ field on an Opportunity record, only managers can edit it. How would you achieve this?

We would create two page layouts. One layout would have the ‘Discount’ field as read-only, while the other would have it editable. Then, I would assign the read-only layout to the sales rep profile and the editable layout to the manager profile.

79.Your organization uses a custom object to track Projects. There’s a request to differentiate between ‘Internal’ and ‘Client’

[Type here]

By Smriti Sharan ([sfdcamlified](#))

projects, capturing different data sets for each. How do you set this up?

I'd start by creating two record types for the Projects custom object: 'Internal' and 'Client'. Next, I'd design two different page layouts, one tailored for internal projects with its specific fields and another for client projects. Each record type will then be associated with its respective page layout. This way, when users create a new project, they can pick the appropriate record type and get the desired page layout to fill in the relevant details.

80.What is the difference between hiding the field from page layout and hiding the field from field level security?

When we hide the field from **page layout** it is just not visible from that page, but it can be visible from reports, search results, list views, related lists, email and mail merge templates, custom links or with API names in the code.

Whereas if we hide the field from **field-level security**, then it is not visible from anywhere.

81. Is it possible to restrict permission for users using permission set?

ANS: No, Permission Set always extends the permission. It does not restrict permission to users.

82.If a user does not have access to a specific record type, will they be able to see the records that have that record type?

Yes, Record type controls only visibility of record on UI but not its access to users. If user does not have access to record type then user will not be able to create records for that record type using UI. But user will be able to see records if they have appropriate permission to do so.

83.In Profile settings, what is difference between “Modify All Data” and “Modify All” ?

Modify All Data : Read, Create, edit, delete, view all and modify all for

[Type here]

By Smriti Sharan ([sfdcamlified](#))

current Profile, regardless of sharing settings.

Modify All : Give Read, Edit, Delete and View All permission to selected Object, Create permission is not included in Modify All permission.

84. A marketing team member can update campaign records (edit) and view existing campaigns (read) but cannot create new campaigns or delete existing ones. Why?

Object-level security controls the permissions users have for specific objects, such as the ability to create, read, edit, and delete records. If a user can read and edit but cannot create or delete records, it indicates that their profile or permission set lacks create and delete permissions for that object.

85. Two users with the same profile can see different sets of records for the same object. What might be causing this discrepancy?

This discrepancy is due to Organization-Wide Defaults (OWD) and other record-level security settings such as role hierarchy, sharing rules, or manual sharing. The OWD settings define the baseline level of access for records, which can be further restricted or extended by these additional mechanisms.

86. User X and User Y have the same profile with edit access to the Account object. The OWD for Accounts is set to private. User X can edit their own Accounts but cannot edit Accounts owned by User Y. Why is this happening?

This is because the OWD is set to private. Even though both users have edit access to the Account object via their profiles, they can only edit the records they own. To allow User X to edit User Y's Accounts, a sharing rule or manual sharing must be implemented.

87. User A is a Sales Rep and User B is their manager. The OWD for Opportunities is set to private. User B can see User A's opportunities, but User A cannot see User B's. Explain why this is the case.

Role hierarchy is in place and is respecting the "Grant Access Using Hierarchies" setting. This setting allows users higher in the hierarchy

[Type here]

By Smriti Sharan ([sfdcamlified](#))

(User B) to see the records owned by users lower in the hierarchy (User A). User A cannot see User B's records because the hierarchy works top-down.

88. The OWD for Cases is private. You need to ensure that all support agents can view and edit each other's cases. How would you configure this in Salesforce?

Create a sharing rule for Cases.

89. User A manually shares an Account record with User B. Later, User A leaves the company and their user account is deactivated. Will User B still have access to the shared Account record?

No, once User A's account is deactivated or ownership of the record changes, the manual sharing settings will be removed, and User B will lose access to the shared Account record.

90. A profile has edit access to the Account object. However, a user with this profile cannot edit the "Annual Revenue" field on the Account. Why might this be the case?

The field-level security settings for the "Annual Revenue" field is set to read-only for the user's profile or permission set. Field-level security overrides object-level permissions.

91. User C needs additional permissions to create Leads, which are not granted by their current profile. How would you provide these additional permissions without changing the profile?

Create a permission set with the necessary Lead creation permissions and assign it to User C. Permission sets are used to extend user access without changing their profile.

92. You need to share Opportunities with specific criteria (e.g., Opportunities where the related Account is in a specific industry) with a group of users. How would you achieve this?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

Implement an Apex sharing rule. Create an Apex trigger or class that checks the criteria (related Account's industry) and programmatically creates OpportunityShare records for the group of users.

93. A custom object "Project" has an OWD set to private. How can you ensure that all users in the "Project Managers" role can read and edit each other's "Project" records?

Create a sharing rule for the "Project" custom object. Set the rule to **share records owned by users in** the "Project Managers" role with the same role, **granting read/write access**.

94. You want to disable the default sharing behavior that allows managers to see their subordinates' records for a custom object. How would you do this?

Uncheck the "Grant Access Using Hierarchies" option for the custom object in the sharing settings. This prevents users higher in the role hierarchy from automatically gaining access to records owned by their subordinates.

95. User D has read access to the Account object but cannot see a related Contact through a lookup field. Why might this be, and how can it be resolved?

Ensure that User D has the appropriate permissions to view the Contact object. Additionally, **check if the Contact is controlled by parent settings**; if so, User D needs access to the parent Account to see the related Contact.

96. How can you track changes made to a field, such as "Annual Revenue," on the Account object?

Enable field history tracking for the "Annual Revenue" field. This will log changes made to the field, including the original and new values, the user who made the changes, and the date of the changes.

97. How can an administrator monitor login attempts to ensure security compliance?

The administrator can review the login history by navigating to the "Login History" section in Salesforce setup. This provides a list of successful and failed login attempts over the past six months.

98. What are the different levels of data access in Salesforce?

[Type here]

By Smriti Sharan ([sfdcamlified](https://sfdcamlified.com))

The levels of data access in Salesforce include Organization Level Security, Object Level Security, Field Level Security, and Record Level Security.

99. What is Organization Level Security?

Organization Level Security controls login access, including password policies, login hours, and IP address restrictions.

100. What is Object Level Security?

Object Level Security determines whether a user can see, create, edit, or delete records of a specific object.

101. What is Field Level Security?

Field Level Security controls whether specific fields on an object are visible and editable for a user.

102. What is Record Level Security?

Record Level Security determines whether a user can see or edit individual records based on ownership and sharing settings such as OWD, role hierarchy, sharing rules, and manual sharing.

101. What is the difference between setting field visibility through page layouts and field-level security?

Setting field visibility through page layouts only affects visibility on that specific layout, while field-level security controls visibility across all parts of Salesforce, including related lists and reports.

102. What is the Setup Audit Trail?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

The Setup Audit Trail logs modifications made to your org's configuration, such as changes to fields, objects, and Apex code.

103. You have ten users needing different levels of access to two apps. Five need access to App A and five need access to both App A and App B. How would you manage this efficiently using profiles and permission sets?

Assign all ten users to the same profile. Create two permission sets: one for access to App A and another for access to both App A and App B. Assign the appropriate permission sets to the respective users.

104. As a system administrator, you are going on vacation and need to grant temporary access to a colleague to manage specific fields. How can you achieve this using Salesforce features?

Create a permission set with access to the specific fields and assign it to your colleague temporarily. Revoke the permission set when you return.

105. How can you ensure that a user cannot see or edit the Opportunity object?

Modify the user's profile to remove read, create, edit, and delete permissions for the Opportunity object.

106. Can you modify standard profiles directly in Salesforce? What is the recommended approach if you need to customize permissions?

Standard profiles cannot be modified directly. The recommended approach is to clone the standard profile and then make the necessary modifications to the cloned profile.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

107.If a user's profile does not grant access to an object, can record-level security settings override this and grant access to specific records?

No, if a user's profile does not grant access to an object, record-level security settings cannot override this. Without access to the object itself, record-level security settings are irrelevant.

108. How do you manually share a record with another user, and what is the purpose of this feature?

To manually share a record, the record owner can go to the record, click on the "Sharing" button, and specify the users or groups to share the record with. This feature allows record owners to grant access to specific users on an as-needed basis.

109.In what scenarios would you use sharing rules instead of manual sharing?

Sharing rules are used when you need to grant access to records based on specific criteria or ownership to a larger group of users automatically. Manual sharing is more suitable for granting access to individual records on a case-by-case basis.

110. What does the "Transfer" permission in the "Public Read/Write/Transfer" OWD setting allow a user to do?

The "Transfer" permission allows users to transfer the ownership of a record to another user. This is in addition to the read and write access provided by the "Public Read/Write" setting.

112.Explain the significance of opportunity and case access settings when creating a new role?

Opportunity and case access settings determine how users in a particular role can interact with opportunity and case records. These

[Type here]

By Smriti Sharan ([sfdcampaified](https://sfdcampaified.com))

settings allow you to specify whether users can view, edit, transfer, or delete these records.

113. How would you write a test class to verify the behaviour of an Apex class with "with sharing"?

In the test class, you would create test data and verify the behavior using `System.runAs()` to simulate different user contexts.

@isTest

```
public class MyClassTest {
```

```
    @isTest
```

```
    static void testWithSharing() {
```

```
        User testUser = [SELECT Id FROM User WHERE Email =  
'test@example.com'];
```

```
        System.runAs(testUser) {
```

```
            MyClass myClass = new MyClass();
```

```
            // assert statements to verify behavior
```

```
        }
```

```
    }
```

114. How can you debug Apex code to see the output of a class running in different sharing modes?

You can use Apex Replay Debugger in VS Code. Set breakpoints, run the debug test, and use step over/into to inspect the outputs line by line in the debug console.

115. What would be the expected result when querying account records in an Apex class defined with "without sharing" versus "with sharing"?

Without Sharing: The query would return all account records in the org, bypassing user permissions.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

With Sharing: The query would return only the account records the current user has access to based on sharing rules.

116. How can you call an inherited sharing class inside a with sharing class, and why might this be necessary?

You can call an inherited sharing class within a with sharing class to ensure the inherited class to ensure the calling class's sharing rules.

```
public with sharing class ParentClass {  
    public void callInheritedClass() {  
        InheritedClass ic = new InheritedClass();  
        ic.someMethod();  
    }  
}
```

117. When writing a test class, why is it important to use the "SeeAllData" annotation, and how does it impact the tests?

Using the "SeeAllData" annotation allows test methods to access all data in the org, not just data created within the test. This is useful for testing behavior with existing records but should be used cautiously as it can lead to unpredictable test results.

118. What will be the result of querying account records in an Apex class marked as "with sharing" if the current user has no permissions to view account records?

If the current user has no permissions to view account records, the query in the "with sharing" class will return zero records.

120. Given a class marked with "with sharing" and a private OWD setting, what records will a user with limited access see when querying the service request object?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

A user with limited access will only see the records they own or have been explicitly shared with them. For example, if the user owns one record and the OWD is set to private, they will only see their own record when querying.

121. How does "inherited sharing" provide flexibility in Apex classes, and what is its primary advantage?

"Inherited sharing" allows an Apex class to adopt the sharing mode of the caller, providing flexibility by ensuring the class respects the caller's sharing context. The primary advantage is that it allows the class to dynamically adjust its sharing behavior based on how it is invoked, enhancing security and consistency.

122. Describe how "inherited sharing" behaves in a parent-child class relationship?

In a parent-child class relationship, if the parent class is marked with a specific sharing mode, the child class with "inherited sharing" will adopt the same sharing mode. This ensures that the sharing settings of the parent class are respected by the child class, maintaining consistent data access control.

123. How does the combination of different sharing modes (e.g., with sharing, without sharing, inherited sharing) affect data access when multiple classes are involved?

- If a class with "with sharing" calls another class with "inherited sharing," the called class will respect user permissions.
- If a class with "without sharing" calls another class with "inherited sharing," the called class will run in system context.
- If both classes use "inherited sharing," the sharing mode will be determined by the initial class that starts the transaction.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

124. Provide an example of when "inherited sharing" would be particularly useful in a Salesforce application?

"Inherited sharing" is useful in scenarios where a class might be invoked from different contexts with varying sharing requirements. For example, a utility class that is used by both a Custom page (requiring user context) and a scheduled batch job (requiring system context) can use "inherited sharing" to dynamically adjust its behavior based on the caller, ensuring appropriate data access in each context.

125. In what scenario might "without sharing" be necessary, and how can you mitigate the risks associated with it?

"Without sharing" might be necessary for administrative operations that require unrestricted access to data, such as data migration or cleanup tasks.

126. How does the with security_enforced keyword affect SOQL queries?

The with security_enforced keyword ensures that the SOQL query respects the field level and object level security settings of the user. If a user does not have access to certain fields, the query will fail with an insufficient permissions error.

127. What are the limitations of using with security_enforced in SOQL queries?

- It cannot be applied to DML statements.
- It is only applicable to fields mentioned in the SELECT and FROM clauses, not in the WHERE clause.
- Traversing polymorphic field relationships is not supported except for Owner, CreatedBy, and LastModifiedBy.
- It only identifies the first error in the SOQL query.
- It throws an exception that developers need to handle.

[Type here]

By Smriti Sharan ([sfdcambified](https://sfdcambified.com))

128. How does the with user_mode keyword improve upon with security_enforced?

The with user_mode keyword can be applied to both SOQL queries and DML operations, ensuring that all fields and objects involved respect the user's permissions. It also applies to fields in the WHERE clause and supports traversing polymorphic field relationships.

129. What is a key feature of the with user_mode keyword in terms of error handling?

The with user_mode keyword identifies all fields that the user does not have access to and captures these in a query exception. Developers can use the queryException.getInaccessibleFields() method to get a list of all inaccessible fields and handle them appropriately.

130. What happens when a user without access to certain fields executes a SOQL query with the with security_enforced` keyword?

The query will fail with an insufficient permissions error. The developer must handle this exception to provide a meaningful message to the user.

131. How do you enforce field level security when performing a DML update operation using with user_mode?

You can enforce field level security by using the database method with the with user_mode keyword. For example:

```
Database.update(records, AccessLevel.USER_MODE);
```

This ensures the update operation respects the user's permissions for the fields involved.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

132. How would you handle a situation where some parts of your Apex code need to enforce field level security while others do not?

For parts of the code that need to enforce security, use with security_enforced or with user_mode. For parts that do not need enforcement, you can either omit these keywords or use with system_mode to explicitly bypass security checks.

133. Why is running Apex in system mode potentially risky?

Running Apex in system mode is risky because it bypasses all user permissions, potentially exposing sensitive data to users who should not have access to it. This can lead to data security breaches and unauthorized access to critical information.

134. How does the Execute Anonymous feature handle sharing rules?

Execute Anonymous always operates in user mode, meaning it enforces sharing rules.

135. If a class with 'with sharing' calls a method in a class with without sharing, what sharing context is enforced?

The method in the class with without sharing will run in the context of with sharing because the calling class enforces sharing rules

136. What is the purpose of the isAccessible method in Apex?

The isAccessible method is used to enforce field and object level security by checking whether the current user has access to create, read, update, or upsert any fields and objects.

137. What are the methods available in the SObjectAccessDecision class, and what do they do?

[Type here]

By Smriti Sharan ([sfdcemplified](#))

`getModifiedIndexes()`: Returns the list of indexes of the records that were modified.

`getRecords()`: Returns the filtered list of records that the user has access to.

`getRemovedFields()`: Returns the list of fields that the user does not have access to.

138. How can you combine `WITH SECURITY_ENFORCED` and `striInaccessible` to enforce security in both SOQL queries and DML operations?

Use `WITH SECURITY_ENFORCED` in SOQL queries to enforce security at the query level and `striInaccessible` before performing DML operations to ensure the user has the necessary permissions:

Example:

```
List<Account> accounts = [SELECT Id, Name, Rating FROM Account  
WITH SECURITY_ENFORCED];
```

```
SObjectAccessDecision decision =  
Security.striInaccessible(AccessType.UPDATABLE, accounts);
```

```
List<Account> filteredAccounts = (List<Account>)decision.getRecords();  
update filteredAccounts;
```

139. What methods can be used to explicitly check user permissions in Apex code?

Methods like `isAccessible`, `isCreateable`, `isUpdateable`, and `isDeletable` from `Schema.DescribeSObjectResult` and `Schema.DescribeFieldResult` can be used to check user permissions.

140. How would you enforce read permission on the Email field of a Contact before querying it?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

```
if (Schema.sObjectType.Contact.fields.Email.isAccessible()) {  
    Contact c = [SELECT Email FROM Contact WHERE Id= :contactId];  
}
```

142. How would you check if the current user has permission to create a Contact with an Email field before performing the operation?

```
if (Schema.sObjectType.Contact.fields.Email.isCreateable()) {  
    Contact c = new Contact(Email = 'test@example.com');  
    insert c;  
}
```

143. How can you ensure that a SOQL query respects the user's field and object permissions?

```
List<Contact> contacts = [SELECT Id, Email FROM Contact WITH  
SECURITY_ENFORCED];  
} catch (Exception e) {  
    // Handle insufficient permission error  
}
```

144. How would you perform a database operation in user mode to enforce user permissions?

```
List<Account> accounts = [SELECT Id, Name FROM Account];  
Database.SaveResult result = Database.insert(accounts,  
Database.UserMode.USER_MODE);
```

145. How can you ensure that records are shared across business units without changing the OWD from private?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

Create sharing rules or use manual sharing to share records with users across business units. If automation is required, use Apex sharing to programmatically share records as needed.

147. What are the advantages of using WITH USER_MODE over WITH SECURITY_ENFORCED in SOQL queries?

- WITH USER_MODE accounts for polymorphic fields like Owner and Task.whatId.
- It processes all clauses in the SOQL SELECT statement, including the WHERE clause.
- It finds all FLS errors in the SOQL query, allowing the use of `getInaccessibleFields()` on `QueryException` to examine the full set of access errors.

Sample Code

```
try {  
    // SOQL query using WITH USER_MODE  
    List<Account> accounts = [SELECT Id, Name, Owner.Name FROM  
Account WITH USER_MODE];  
  
    // Process the retrieved accounts  
    for (Account acc : accounts) {  
        System.debug('Account Name: ' + acc.Name + ', Owner Name: '  
+ acc.Owner.Name);  
    }  
} catch (QueryException e) {  
    // Handle query exceptions and inaccessible fields  
    System.debug('QueryException: ' + e.getMessage());  
    List<String> inaccessibleFields = e.getInaccessibleFields();  
    if (!inaccessibleFields.isEmpty()) {  
        System.debug('Inaccessible Fields: ' + String.join(', ',  
inaccessibleFields));  
    }  
}
```

148. Explain the difference between implicit and explicit sharing in Salesforce.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

Implicit Sharing: Automatically granted by Salesforce based on ownership, role hierarchy, and sharing rules. This type of sharing cannot be modified directly.

Explicit Sharing: Manually granted by users or programmatically via Apex managed sharing. It provides specific access to records beyond the default sharing settings.

149. What are the key properties of a share object in Salesforce?

objectNameAccessLevel: The level of access granted (Edit, Read, All).

ParentId: The ID of the record being shared.

RowCause: The reason for granting access.

UserOrGroupId: The ID of the user or group being granted access.

150. What is the RowCause field in the context of sharing objects and what are some possible values?

The RowCause field indicates the reason for granting access. Some possible values are:

- ImplicitChild
- ImplicitParent
- Owner
- Team
- Rule
- TerritoryRule
- Manual
- TerritoryManual

150. How would you programmatically create a manual share for a custom object called Job to grant read access to a specific user?

```
public class JobSharing {
```

[Type here]

By Smriti Sharan ([sfdcamlified](#))

```
public static boolean manualShareRead(Id recordId, Id
userOrGroupId){
    Job__Share jobShr = new Job__Share();
    jobShr.ParentId = recordId;
    jobShr.UserOrGroupId = userOrGroupId;
    jobShr.AccessLevel = 'Read';
    jobShr.RowCause = Schema.Job__Share.RowCause.Manual;
    Database.SaveResult sr = Database.insert(jobShr, false);
    if(sr.isSuccess()){
        return true;
    } else {
        Database.Error err = sr.getErrors()[0];
        if(err.getStatusCode() ==
StatusCode.FIELD_FILTER_VALIDATION_EXCEPTION &&
err.getMessage().contains('AccessLevel')){
            return true;
        } else {
            return false;
        }
    }
}
```

151. What are the considerations when creating Apex managed sharing for Customer Community Plus users?

Share objects such as AccountShare and ContactShare aren't available to Customer Community Plus users. To enable sharing, use triggers that operate with the without sharing keyword or use an inner class with the same keyword to perform the DML operation.

152. Give a scenario where you would use apex sharing?

There is a custom object called Project__c. The business requirement is that when a project is marked as "In Progress," it should be shared with the project manager. When the project is marked as "Completed," it should be shared with the account manager. This logic is dynamic and depends on the status of the project.

153. When to Use Apex Managed Sharing Recalculation?

[Type here]

By Smriti Sharan ([sfdcamlified](#))

Organization-Wide Default Changes:

When the organization-wide sharing default (OWD) access level for an object changes, Salesforce automatically recalculates sharing for all records on that object. Apex managed sharing recalculations can ensure that specific sharing rules are re-applied correctly.

Error Handling and Recovery:

When a locking issue or other error prevents the application of sharing rules, recalculating Apex managed sharing can help reapply those rules.

Dynamic Sharing Requirements:

When business logic requires dynamic and complex sharing that cannot be handled by standard sharing rules alone.

154. When is Apex managed sharing recalculated automatically by Salesforce?

Salesforce automatically recalculates sharing for all records on an object when its organization-wide sharing default access level changes.

155. What must an Apex class implement to perform Apex managed sharing recalculations?

An Apex class must implement the Database.Batchable interface to perform Apex managed sharing recalculations.

156. How can administrators monitor or stop the execution of an Apex recalculation?

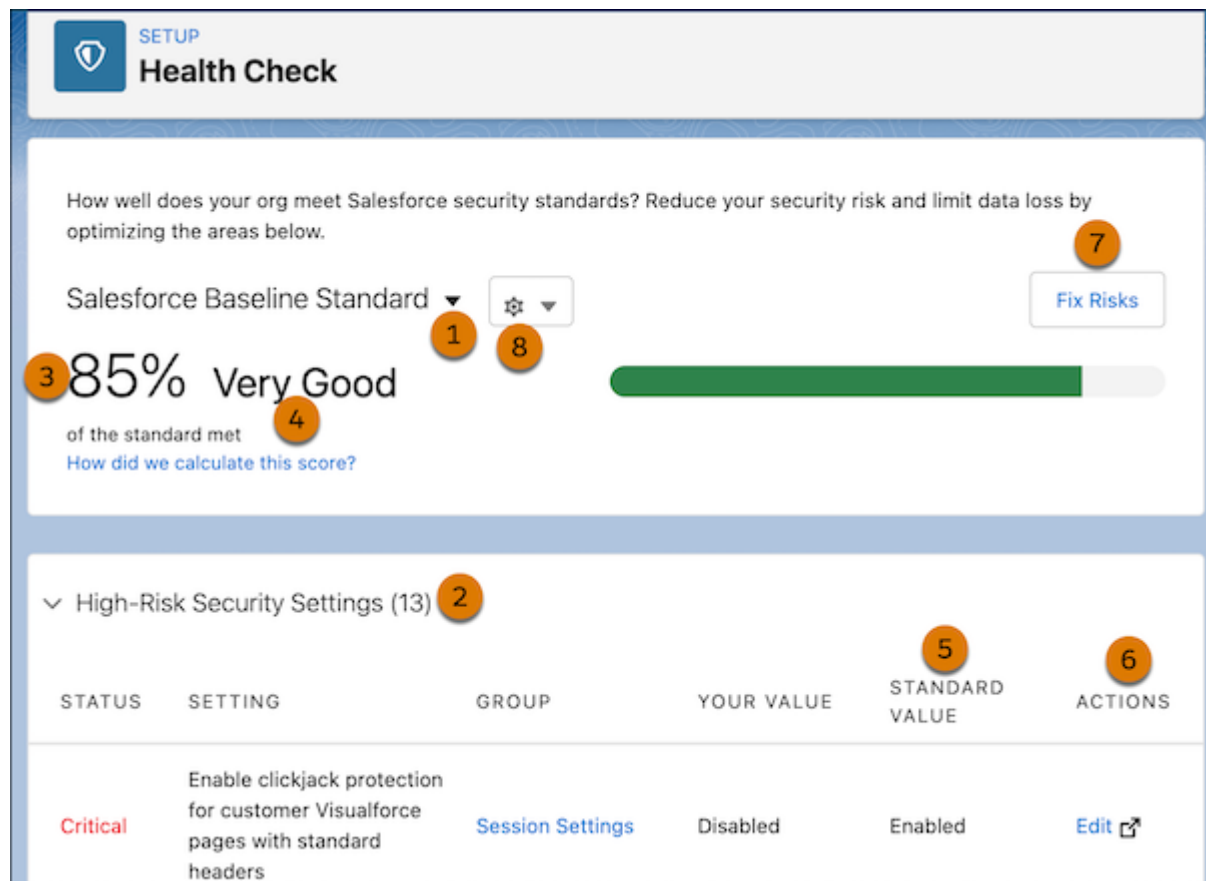
Administrators can monitor or stop the execution of an Apex recalculation from Setup by entering "Apex Jobs" in the Quick Find box and selecting "Apex Jobs".

[Type here]

By Smriti Sharan ([sfdcamlified](#))

157.What is the purpose of Salesforce Health Check?

The purpose of Salesforce Health Check is to identify **and fix potential vulnerabilities** in your security settings from a single page. It provides a **summary score showing how your organization measures against a security baseline**, such as the Salesforce Baseline Standard or custom baselines.



158. User A can see and edit the Email field on the Contact object, but User B cannot see this field at all?

User B's profile does not have Read access for the Email field on the Contact object.

[Type here]

By Smriti Sharan ([sfdcamlified](#))

159. User A, a manager, can see all the records owned by User B, a subordinate. However, User B cannot see any records owned by User A. What should we do so User B can see User A record?

The role hierarchy allows upward visibility (managers see subordinates' records) but not downward. Create a sharing rule that grants User B access to the records owned by User A.

160. The Case object is set to Private. User A shares a case manually with User B, but User B still cannot see the case. What could be the issue, and how would you fix it?

User B may not have the necessary object-level permissions to view cases. Ensure User B's profile has at least Read access to the Case object. Verify the manual sharing settings and correct any errors.