Ankita

BASIC SALESFORCE ADMIN INTERVIEW QUESTIONS (MUST KNOW)

1. Record Sharing and Security

Q1: What is the difference between public and private record sharing in Salesforce?
A1:

- **Public**: All users can see the records based on their profile permissions and sharing rules. For example, in a public sharing model, all users can see all records unless specific sharing rules restrict it.
- **Private**: Records are visible only to the record owner and those with specific sharing permissions.

Q2: What is a sharing rule?
A2:
Sharing rules are used to give specific users access to records that they might not normally be able to access. These are defined at the organization level and can be based on criteria or ownership.

2. Custom Settings and Custom Metadata

Q1: What is the difference between Custom Settings and Custom Metadata Types?
A1:

- **Custom Settings**: They are similar to custom objects, but their data is cached and can be accessed without making a query, making them faster. They are typically used for configuration or data that does not change frequently.
- **Custom Metadata Types**: They are also like custom objects but are deployable and upgradable. They allow users to create reusable metadata configurations across environments.

Q2: How do you access custom metadata in Apex?
A2:
You can query custom metadata in Apex using the SOQL query syntax like this:

```
Apex

List<CustomMetadataType__mdt> metadataRecords = [SELECT MasterLabel, Field__c
FROM CustomMetadataType__mdt];
```

3. Record Types, Layouts, and Dynamic Forms

Q1: What are Record Types in Salesforce?
A1:
Record Types allow you to offer different business processes, picklist values, and page layouts to different users based on their profile. For example, you can have different page layouts for an Opportunity depending on whether it's a "New Business" or "Renewal."

Q2: What is the difference between Page Layout and Dynamic Form?
A2:

- **Page Layout**: Controls which fields, sections, buttons, and links appear on the page, and the layout of those elements.
- **Dynamic Form**: A more flexible approach for customizing record pages, allowing users to conditionally display fields and sections based on specific criteria without using code.

4. User Management

Q1: What is the difference between Profile and Permission Set?
A1:

- **Profile**: Defines the base permissions (like CRUD access to objects, fields, and other features) for users. A user can only have one profile.
- **Permission Set**: Provides additional permissions to users on top of their profile without changing the profile itself. A user can have multiple permission sets.

Q2: What are User Licenses in Salesforce?
A2:
User licenses define the baseline features and functionality available to a user, such as Salesforce, Salesforce Platform, Chatter Free, etc. It determines what the user can do in Salesforce, based on the features and access assigned.

5. Freeze vs Deactivating the User

Q1: What is the difference between freezing and deactivating a user?
A1:

- **Freezing a User**: Temporarily prevents a user from logging into Salesforce but retains their permissions and data access.
- **Deactivating a User**: Fully disables the user, preventing access to the organization. A deactivated user can't log in, and their data access is also removed unless shared with another active user.

6. Scoping Rules and Duplicate Rules

Q1: What are Duplicate Rules in Salesforce?
A1:
Duplicate Rules help prevent the creation of duplicate records by defining criteria for identifying duplicates and setting actions to take when a duplicate is found (such as blocking or alerting the user).

Q2: What are Scoping Rules in Duplicate Rules?
A2:
Scoping rules define the data sets that are checked for duplicates. For example, a scoping rule might only check for duplicates within a particular account, region, or record type.

7. Relationships & Data Models

Q1: What are Lookup, Master-Detail, and Junction Relationships in Salesforce?
A1:

- **Lookup Relationship**: A one-to-one relationship where one object can reference another object.
- **Master-Detail Relationship**: A tightly bound one-to-many relationship where the child record's lifecycle is tied to the parent.
- **Junction Relationship**: A many-to-many relationship created by using two master-detail relationships.

Q2: What is a Self Lookup Relationship?
A2:
A self-lookup relationship occurs when an object has a lookup field that references itself, allowing you to link a record to another record of the same object.

8. Reports and Dashboards

Q1: What is the difference between a Report and a Dashboard in Salesforce?
A1:

- **Reports**: Are the raw data in tabular, summary, matrix, or joined format. They allow users to filter, group, and summarize data.
- **Dashboards**: Provide a visual representation of multiple reports, allowing users to track key metrics at a glance with graphs, charts, and tables.

9. Approval Process

Q1: What are the types of approval processes in Salesforce?
A1:

- **Standard Approval Process**: A set of predefined steps and rules for approvals, such as a manager reviewing and approving an opportunity.
- **Custom Approval Process**: Tailored to a company's needs, with customizable rules, actions, and steps.

Q2: How many steps can an approval process have?
A2:
An approval process can have multiple steps, with each step corresponding to a stage of the approval cycle, such as "Submit for approval," "Manager Approval," and "Final Approval." There's no hard limit, but a process typically has 1 to 10 steps.

10. Validation Rules

Q1: What is a Validation Rule in Salesforce?
A1:
A validation rule ensures data entered into Salesforce meets specified criteria, preventing users from saving records with invalid data. It returns an error message if the data doesn't meet the conditions.

Q2: What are the common use cases for Validation Rules?
A2:

- Ensuring required fields are not left blank.
- Validating that email addresses follow the proper format.
- Ensuring the start date is before the end date on opportunities or contracts.

11. Public Groups, Queues, and List Views

Q1: What is a Public Group in Salesforce?
A1:
A public group is a set of users that can be used in sharing rules, reports, and dashboards to grant access to records. It is a collection of individual users, roles, or other groups.

Q2: What is the difference between Queues and List Views?
A1:

- **Queues**: Used to manage records (e.g., cases, leads) that need to be worked on by a group of users. Records in a queue are assigned to specific users for processing.
- **List Views**: Provide a way to filter and display a set of records based on user-defined criteria. List views are for displaying data in a table format.

Project Based Explanation:

1. Record Sharing and Security

Q1: How would you implement record-level security in a multi-departmental Salesforce org?
A1:

- Implement **Profiles** to control access to objects and fields based on department roles.
- Use **Sharing Rules** to allow different departments to access relevant records without opening everything to all users.
- Use **Role Hierarchy** to ensure that higher-level users can access records owned by lower-level users in the same department.
- **Field-Level Security** should be used to hide sensitive information from certain users based on their department.
- For more fine-grained control, **Apex Managed Sharing** can be used when the sharing needs are more complex.

2. Custom Settings and Custom Metadata

Q1: How would you use Custom Metadata to manage configuration in a project?
A1:
For a project where you need reusable, deployable configurations across multiple environments (e.g., sandbox, production):

- Create **Custom Metadata Types** for each configuration requirement.
- Store values like thresholds, approval settings, or external system API keys in custom metadata.
- Query and apply custom metadata values in Apex or Lightning Components to customize functionality dynamically without needing hardcoded settings.
- Deploy Custom Metadata records from one environment to another using change sets or Salesforce DX.

Q2: Can you use Custom Settings for storing per-user configurations in a project?
A2:
Yes, **Custom Settings** are ideal for storing per-user or per-profile configurations that might not need to be deployed. You can use **List Custom Settings** for application-wide configurations or **Hierarchy Custom Settings** to store data specific to individual users or profiles, improving performance by caching data.

3. Record Types, Layouts, and Dynamic Forms

Q1: How would you implement dynamic page layouts for users in a multi-stage approval process?
A1:

- **Record Types** can be used to differentiate between different stages of the approval process (e.g., "Draft," "In Review," "Approved").
- For each **Record Type**, assign unique **Page Layouts** that show relevant fields at each stage.
- Use **Dynamic Forms** to conditionally show or hide fields based on record values or user inputs, making the form more efficient for users.
- You can also set visibility rules for specific fields based on conditions, allowing users to only see fields that are relevant at their stage in the approval process.

4. User Management

Q1: How do you manage user access and permissions across different departments in a large Salesforce project?
A1:

- **Profiles** should be defined for each department or job function, ensuring that each user only has access to the objects, fields, and data they need.
- **Permission Sets** can be used to assign additional permissions to users who require access to extra features without modifying their profiles.
- **Role Hierarchy** ensures that managers or supervisors can access records owned by their subordinates.
- Use **Public Groups** and **Sharing Rules** for cross-departmental collaboration, ensuring that users from different departments can access necessary records based on shared criteria.
- When new users are onboarded, a standard **User Setup Process** should be followed to assign them the appropriate profile, role, and permissions.

5. Freeze vs Deactivating the User

Q1: When implementing a user deactivation process, how would you ensure there's minimal disruption to business operations?
A1:

- First, **freeze** the user to prevent login and check if their existing tasks and records need reassignment.
- Reassign any open tasks or active records (e.g., opportunities, cases) to another active user or queue before deactivating the user.
- Deactivate the user only after ensuring all records are transferred and the user's workflow is closed.
- Set up proper **email notifications** or alerts when a user is frozen or deactivated to ensure other users are aware of the change.

6. Scoping Rules and Duplicate Rules

Q1: How would you implement duplicate management in Salesforce for a project involving leads from different channels?
A1:

- Define **Duplicate Rules** based on the lead data (e.g., email, phone number). Use **Matching Rules** to determine when a lead is considered a duplicate (e.g., same email or phone number).
- Set the scope for checking duplicates across different lead source channels using **Scoping Rules**, such as checking duplicates only within the same lead source (e.g., Web Leads).
- If a duplicate is found, use actions like **Block** to prevent entry or **Alert** to notify the user of the duplicate.
- Regularly monitor and refine these rules to improve accuracy as more data gets accumulated in the system.

7. Relationships & Data Models

Q1: How would you design a data model for a project where each product can have multiple related documents and tags?
A1:

- Use a **Master-Detail Relationship** between Product and Document to ensure that if a Product is deleted, all associated documents are also deleted.
- For tags, use a **Junction Object** to create a many-to-many relationship between Product and Tag. This allows products to be associated with multiple tags, and tags to be reused across different products.
- **Lookup relationships** can also be used when one object should reference another object without affecting the record lifecycle.

8. Reports and Dashboards

Q1: How would you implement a real-time sales dashboard for a project to track opportunities and deals?
A1:

- Create **Reports** that track opportunity stages, revenue, and win rates across various stages.
- Use **Dynamic Dashboards** to provide real-time data visualization, showing charts, graphs, and key performance indicators (KPIs) such as conversion rate, total sales, and forecasted revenue.
- Allow users to filter the data based on specific regions, product lines, or sales reps using **Interactive Filters**.
- For effective real-time tracking, ensure that the data is updated frequently and that users are alerted of any significant changes or opportunities through **Reports Subscriptions**.

9. Approval Process

Q1: How would you design an approval process for a project involving complex workflows, such as multi-level approvals and conditional steps?
A1:

- Start with creating a **Custom Approval Process** that defines the stages (e.g., "Initial Approval," "Manager Approval," "Final Approval").
- Use **Entry Criteria** to define which records enter the approval process based on conditions (e.g., Opportunity Value > $10,000).
- Implement **Dynamic Approval Steps** where certain steps are skipped or added based on record field values or user roles (e.g., an executive might approve in place of a manager for high-value opportunities).
- Use **Automated Actions** such as email notifications and field updates to move the approval forward and notify users.

10. Validation Rules

Q1: How would you implement validation rules to ensure data quality in a Salesforce project?
A1:

- Create **Validation Rules** to enforce correct data entry, such as making sure the **Close Date** on an Opportunity is not earlier than the **Created Date**.
- Use formulas to restrict certain values, like preventing a discount greater than a predefined threshold for certain products.
- For projects where users may input complex data, consider creating a **custom error message** to make the validation feedback more helpful and user-friendly.
- Test all validation rules in a sandbox environment before deploying to ensure they don’t impact workflows negatively.

11. Public Groups, Queues, and List Views

Q1: How would you implement a case management solution using Queues and List Views for a customer service project?
A1:

- Use **Queues** to manage cases that need attention from customer service representatives. Assign cases to specific queues based on criteria (e.g., high-priority cases go into a "High Priority" queue).
- Create **List Views** that filter and display cases based on important factors such as priority, status, or the case owner.
- Set up **Public Groups** to share case access among relevant teams (e.g., sales, support) who may need to collaborate.
- Implement **Escalation Rules** to ensure cases that haven’t been resolved in a timely manner are automatically reassigned to higher-level queues or managers.