Salesforce Admin/ Developer Questions

Salesforce Techbook- Online Mock interviews
theOnlineSupport_Salesforce- Interview questions

Permission set:

A permission set is a collection of settings and permissions that give users access to various tools and functions.
Permission sets extend users' functional access without changing their profiles.

Profiles and permission sets both control CRED (Create, Read, Edit, Delete) permissions on Objects, fields, user settings,
tab settings, app settings, Apex class access, Visualforce page access, page layouts, record types, login hours and login IP ranges.

Essentially, a user's profile is the baseline authorization of access to the Org. Permission sets are, as the name implies, a set of additional CRED permissions that can be applied to different profiles. Typically they are task-based and related to different Objects and managed packages.

For example, Sales users may be assigned a permission set giving them access to a CPQ app to generate quotes

Types of profiles:

Standard , Custom

We cannot delete standard ones.
Read Only, Standard User, Marketing User, Contract Manager, Solution Manager & System Administrator.
Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

Custom ones defined by us. They can be deleted if there are no users assigned to that particular one.

Handler class vs Helper class

The term Handler is basically used while using triggers. Handlers classes are used to Handle the execution of triggers.
If you are working on custom application development on force.com or customizing salesforce to do event-driven updates,
you will end up writing quite a few triggers and in some cases more than one trigger on the same object.

You usually deal with helper classes when you are working on the Lightning platform.
The helper is JavaScript "shared code", and reduces the memory footprint of each component by sharing common methods.
The controller itself is duplicated each time, so it should be as small as possible to reduce memory usage.

Why we don't use multiple triggers on one Object?

Writing multiple triggers on Single Object renders the system unable to recognize the Order of execution of trigger.

Moreover, each trigger that is invoked does not get its own governor limits. Instead, all code that is processed.

Bulkyfing the Code:

Bulkifying your code refers to combining repetitive tasks in Salesforce Apex such that the code properly handles more than one record at a time

With / Without Sharing:

Important -

> if a method is defined in a class declared with 'with sharing' is called by a class declared with 'without sharing', the method will execute with sharing rules enforced. The class doesn't enforce sharing rules except if it acquires sharing rules from another class. Ex. Class A (with sharing) calls a method from Class B(Without sharing) then complete context is 'with sharing'
> Inner classes do not inherit the sharing setting from their container class.

System context means Without sharing
User context means with Sharing

Based on the context, it can pull the data

Inherited Sharing: With inherited sharing , we can run our apex code with both With or without sharing settings, depending on the context it is being called from.

Example:

We have the data of Mumbai and Maharashtra.

Mumbai data can be pulled from inner class and Maharashtra data can be called from Outer class.

Scenario:

```
Public Without sharing class Outer{
//Outer class code//
without sharing class inner {
//Inner class code//
}
}
```

Ans: In this scenario only data from Mumbai can be called due to sharing rules implements on inner class code

Flows:

Will all editions supports flows in salesforce: No
They are supported inn Enterprise , Performance and Unlimited only

What happens when you delete a flow from your org?
It will permanently deleted and cannot be recovered

What happens when user navigates back to previous screen while filling out values on multiscreen flow?

It lost the current data that entered

What data types restrictions are there when sending information from one screen to another?

Lists, sets and Map.

We can use string, integer, decimals, Boolean , dates and datetimes.

Real time example where you used flows to solve customer problems?

I have used flows in a few different ways to help my customers. one example is When I built a flow that helped a customer automatically create tasks for their Sales reps whenever a new lead came in. This helped the customer keep track of their sales reps progress and make sure that nothing fell through the cracks. Another ex is when I built a flow that helped a customer automatically generate quotes for their customers. this helped the customer save time on quote creation and it also helped ensure that all necessary information was included in the quote

Order of execution of trigger :

Before save record triggered flow
Before save apex triggers
after save apex trigger
after save record trigger flow

Best practices of flow:

Naming Convention of the flow
Always test your flows
Consider using sub flows
Never perform DML Statements in Loops
Document your flows
Never Hardcode ids
Plan for faults
Utilize before-save flows for Same record updates
use after save flow instead of process builder/workflow
Flow isn't always the best idea

Salesforce default feature is convert lead into account , contact , opportunity.

What is the necessity of Custom object instead of standard custom.

If it has its own limitation to fulfill your requirement then we use standard otherwise custom object is used

What is master detail relationship?

We need to create 2 MD relationship fields on object. Whenever a parent record is deleted , automatically delete child records as well.

Example:

How many rollup summary fields can we create on objects:
25

Why do we use Custom settings over custom Objects?

Custom setting in Salesforce are a way to store custom configuration data that can be accessed and referred by apex code or formulas.

They are Similar to custom objects in that they are allow you to store data but they are designed to be used for specific use cases such as storing application configuration data or providing default values for custom fields.

Advantages over custom objects:

-These are cached, which means that they are stored in application's memory and can be accessed quickly. This can improve performance for frequently accessed data.
-These are used in formulas, Validation rules and workflow rules which allow you to use the data stored in them to make decisions about how the system behaves.
-These are available in both apex and visualforce , which allows you to reference the data stored in them from both server-side and client-side code.

These have built-in support for hierarchical data, which allows you to store data at the organization, profile, or user level. This can be useful for storing data that needs to be different for different users or profiles.

Custom settings are best when you need to store data that needs to be accessed frequently and needs to be used in formulas, Validation rules and workflows. They also useful when you need to store data that needs to be different for different users or profiles, or when you need to reference the data stored in them from both server side and client side.

Types of Custom settings:

- List
- hierarchal

Salesforce access management/ User Management:

- Allowing only authorized users to access salesforce
- Setting password Policies
- Restricting IP ranges for Users
- Restricting login hours for users

Data Security in Salesforce:

It deals with the security or Sharing settings of data and visibility between users or groups of users across the organization

Force.com provides a flexible, layered sharing model that makes it easy to assign different data sets to different sets of users

Security and sharing model can be configured entirely

using the user interface yet it is implemented at the api level

which means

any permissions specified for objects, records and fields apply even if a user query or update the data via api calls

Level of Data Access:

- Organization Level
- Object Level
- Field Level
- Record Level

Object Level security in Salesforce:

Salesforce Object Level Security provides the simplest way to control data access. It prevents a user or group of users from creating, viewing, editing, or deleting any records of an object by setting permissions on that object.

There are two ways of setting object permissions:

- Profiles
- Permission Sets

Profiles:

It determines the objects a user can access and the permissions a user has on any object record.

A profile is a collection of settings and permissions that determine which data and features in the platform users have access to.

Permission sets:

It provides additional permissions and access settings to users.

Use permission sets to grant additional access to specific users on top of their existing profile permissions, without having to modify existing profiles, create new profiles or grant an administrator profile where it’s not necessary.

There are a couple of ways to use permission sets:

1. To grant access to custom objects or entire apps.
2. To grant permissions-temporarily or long term-to specific fields

Organizational- Wide default Settings:

Organization-Wide default or Organization-Wide sharing settings determine the baseline level of access for all records of an object. Organization-wide defaults can never grant users more access than they have through their object permissions

- Public read/ Write
- Public read only

- Private

How to restrict users to access another user’s records?

OWD

Sharing Rules

Role Hierarchy

Manual Sharing

Record ID: Each record in salesforce system has a Unique ID which is known as Record ID. It is a System generated and cannot be edited or deleted. It is created every time when a new record is created.

Record Type: It allows us to specify a category of records that display different picklist values and Page layouts

Admins can associate record types with profiles so that different users should see different picklist values and Page Layouts in the record details page

Record Owner: Every record belongs to its particular owner. When a record in an Object is created , an Owner is automatically assigned to the record who has all the rights of that record

By Default, the user who created the that record is the owner of that record.

How to change the Ownership of the record?

You can give ownership of a record to another user as long as that user has at least Read permission for the type of record being transferred.

Difference between Helper and Controller in Aura Component:

The helper is designed to have shared code in it. So, you could have code shared by your renderer and your controller or by multiple controller functions. That should go in the helper JS file.

Helper code can be shared among components when they are related through inheritance. If one component extends another it inherits its super component's helper and can override it.

Events: A component registers that it may fire an event in its markup. Events are fired from JavaScript controller actions that are typically triggered by a user interacting with the user interface.

There are two types of events in the framework:

- **Component events** are handled by the component itself or a component that instantiates or contains the component.
- **Application events** are handled by all components that are listening to the event. These events are essentially a traditional publish-subscribe model.

The Developer Console enables you to perform these functions.

- Use the menu bar (1) to create or open these Lightning resources.

- Application
- Component
- Interface
- Event
- Tokens

- Use the workspace (2) to work on your Lightning resources.
- Use the sidebar (3) to create or open client-side resources that are part of a specific component bundle.
  - Controller
  - Helper
  - Style
  - Documentation
  - Renderer
  - Design
  - SVG

Access Modifiers/Specifiers:

Apex allows you to use the private, protected, public, and global access modifiers when defining methods and variables.

private

This access modifier is the default, and means that the method or variable is accessible only within the Apex class in which it's defined.

public

This means that the method or variable is accessible by all Apex within a specific package.

global

This means the method or variable can be used by any Apex code that has access to the class, not just the Apex code in the same application

W**hat is the Component Reference is used to make table in aura:** Using Datatable

What is the component reference is used to make structured data in a table: Using Tree Grid

Aura Component Bundle:

L**ightning Tab**	implements="force:appHostable"	Creates a component for use as a navigation

		element in Lightning Experience or Salesforce mobile apps.
Lightning Page	implements="flexipage:availableForAllPageTypes" and access="global"	Creates a component for use in Lightning pages or the Lightning App Builder.
Lightning Record Page	implements="flexipage:availableForRecordHome, force:hasRecordId" and access="global"	Creates a component for use on a record home page in Lightning Experience.
Experience Builder Site Page (previously Lightning Communities Page)	implements="forceCommunity:availableForAllPageTypes" and access="global"	Creates a component that's available for drag and drop in the Experience Builder.
Lightning Quick Action	implements="force:lightningQuickAction"	Creates a component that can be used with a Lightning quick action.
	Lightning application bundle	
Lightning Out Dependency App	extends="ltng:outApp"	Creates an empty Lightning Out dependency app.

Difference between and <Span> tag?

 is a block level element and <Span> is a inline element

What is the Use of Controller in Salesforce Aura Components?

Controller– A JavaScript Controller is used for handling client-side processing or an apex controller is preferred for server-side processing

Use of @Auraenabled :

It is used to access apex methods and its properties

How to pass data into Apex controller:

• Use **action.setParams()** in JavaScript to set data to pass to an Apex controller.

Use **action.setcallback()** is used to make callback that is executed after a server side actions

We can get the return data from server side using return statement

response.getState() gets the state of the action returned from the server.

$A.enqueueAction(action); is used to add server side actions into a queue

Different Data Types in Aura:

Putboolean,Putint,Putlong,Putdecimal,putstring,Putblob,putdatetime

Blob: Blob is used for holding Binary Data values like 01,101,0010 etc

Data Type: it defines what type of value that a variable is holding

Variable: It is used to store data.

Collection data Types: List, Set ,Map

List is a ordered collection and allows duplicates values

Set is a Unorder collection and doesn't allow duplicate values

Map contains a Key-Value Pair. Each Key is associated with a Value

Assignment rules and escalation rules:

Assignment rules refer to a lead or case that is assigned based on criteria where as Escalation rules is defined to period of time to solve the issue.

Validation rules:

It is used to validate the record before or after the record gets saved

Related List - Single component ***shows a list of related records based on one specific object***

Converting a webpage into pdf using visualforce?

```
<apex:page renderAs="pdf">
```