Salesforce Developer Interview Questions & Answers For 2+ Years Experience

Created By: Prem Singh Foujdar
Sr.Salesforce Developer

SALESFORCE BASICS

Q1: What is Salesforce? Ans: Salesforce is a cloud-based CRM (Customer Relationship Management) platform that helps companies manage their customers, sales, and services. It runs on the internet, so you don't need to install anything on your computer.

Q2: What is CRM? Ans: CRM means Customer Relationship Management. It's a system that helps companies manage all their interactions with customers and potential customers. It stores customer information, tracks sales, and helps provide better service.

Q3: What are the different Salesforce clouds? Ans: Main clouds are:

- Sales Cloud - for sales teams
- Service Cloud - for customer support teams
- Marketing Cloud - for marketing teams
- Commerce Cloud - for online shopping
- Community Cloud - for creating portals

Q4: What is the difference between Sales Cloud and Service Cloud? Ans: Sales Cloud helps sales teams sell products (leads, opportunities, quotes). Service Cloud helps support teams help customers (cases, solutions, knowledge base).

SALESFORCE OBJECTS & DATA MODEL

Q5: What is an Object in Salesforce? Ans: An Object is like a database table. It stores specific types of information. For example, Account object stores company information, Contact object stores people information.

Q6: What are Standard Objects? Ans: Standard Objects are pre-built objects that come with Salesforce. Examples: Account, Contact, Lead, Opportunity, Case, Campaign.

Q7: What are Custom Objects? Ans: Custom Objects are objects that you create yourself based on your business needs. They end with "__c". Example: Student__c, Invoice__c.

Q8: What is the difference between a Role and a Profile? Ans: Profile controls WHAT you can do (permissions like create, read, edit, delete). Role controls WHAT records you can see (record-level access through role hierarchy).

Q9: What is a Record Type? Ans: Record Type allows you to offer different page layouts, picklist values, and business processes to different users on the same object. Example: Different types of Opportunities like "New Business" and "Renewal".

Q10: What are the types of relationships in Salesforce? Ans:

- Lookup Relationship - simple link between two objects (not required)
- Master-Detail Relationship - strong link, child depends on parent
- Many-to-Many Relationship - created using Junction Object
- Self Relationship - object relates to itself

Q11: What is the difference between Lookup and Master-Detail relationship? Ans:

- Lookup: Child can exist without parent, no sharing, no rollup summary
- Master-Detail: Child cannot exist without parent, inherits sharing, can have rollup summary fields

Q12: What is a Junction Object? Ans: Junction Object is used to create Many-to-Many relationships. It has two Master-Detail relationships connecting two parent objects. Example: Student__c and Course__c connected by Enrollment__c.

Q13: What is a Roll-Up Summary field? Ans: It's a field on master object that calculates values from child records. Operations: SUM, COUNT, MIN, MAX. Example: Total Amount of all Opportunities on an Account.

SALESFORCE SECURITY

Q14: What are Organization-Wide Defaults (OWD)? Ans: OWD is the baseline level of access users have to each other's records. Options: Private, Public Read Only, Public Read/Write. It's the most restrictive setting.

Q15: What is a Profile? Ans: Profile is a collection of settings and permissions that determine what a user can do in Salesforce. It controls object permissions, field permissions, tab settings, and app settings.

Q16: What is a Permission Set? Ans: Permission Set gives additional permissions to users without changing their profile. It extends user access. You can assign multiple permission sets to one user.

Q17: What is the difference between Profile and Permission Set? Ans: Profile is assigned to every user (only one per user) and defines baseline permissions. Permission Set provides additional permissions on top of profile (multiple per user).

Q18: What is Sharing Rule? Ans: Sharing Rules are used to extend access to users beyond OWD. They allow you to share records with specific groups of users. Example: Share all Western Region accounts with Sales Managers.

Q19: What is Manual Sharing? Ans: Manual Sharing allows record owners to give specific users or groups access to their records manually, one at a time.

Q20: What is Field-Level Security? Ans: Field-Level Security controls whether users can see and edit specific fields on an object. You can set fields as Visible, Read-Only, or Hidden for different profiles.

APEX BASICS

Q21: What is Apex? Ans: Apex is Salesforce's programming language. It's similar to Java and runs on Salesforce servers. We use it to write business logic and customize Salesforce beyond clicks.

Q22: Is Apex case-sensitive? Ans: No, Apex is NOT case-sensitive for keywords and variable names. But SOQL is case-sensitive for field values.

Q23: What are the different types of collections in Apex? Ans:

- List - ordered collection, allows duplicates
- Set - unordered collection, no duplicates
- Map - key-value pairs, keys must be unique

Q24: What is the difference between List and Set? Ans: List maintains order and allows duplicate values. Set doesn't maintain order and only stores unique values.

Q25: What are Apex data types? Ans:

- Primitive types: Integer, Long, Double, Decimal, String, Boolean, Date, Datetime, Time, ID
- Complex types: sObject, List, Set, Map, Enum

Q26: What is sObject? Ans: sObject is any object that can be stored in Salesforce database. Both standard and custom objects are sObjects. Example: Account, Contact, Student__c.

Q27: What are Apex classes? Ans: Apex Class is a template/blueprint that contains variables and methods. It's like a container for your code.

Q28: What are access modifiers in Apex? Ans:

- Private - only accessible within the same class
- Public - accessible by any Apex in the same application
- Global - accessible by any Apex code in any application

- Protected - accessible within the class and its subclasses

Q29: What is a Constructor? Ans: Constructor is a special method that runs automatically when you create an object of a class. It has the same name as the class and no return type.

Q30: What is method overloading? Ans: Method overloading means having multiple methods with the same name but different parameters (different number or type). Example: add(int a, int b) and add(int a, int b, int c).

APEX TRIGGERS

Q31: What is a Trigger? Ans: Trigger is Apex code that executes before or after specific events on records (insert, update, delete, undelete). It's used for automation.

Q32: What are the types of Triggers? Ans:

- Before Triggers - run before record is saved to database
- After Triggers - run after record is saved to database

Q33: What are Trigger events? Ans:

- before insert, after insert
- before update, after update
- before delete, after delete
- after undelete

Q34: What are Trigger context variables? Ans:

- Trigger.new - list of new versions of records
- Trigger.old - list of old versions of records
- Trigger.newMap - map of new records with IDs
- Trigger.oldMap - map of old records with IDs
- Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete
- Trigger.isBefore, Trigger.isAfter
- Trigger.size - number of records

Q35: When to use Before Trigger and After Trigger? Ans:

- Before Trigger: When you want to update or validate the same record before saving
- After Trigger: When you want to update related records or when record ID is needed

Q36: What is the best practice for Triggers? Ans:

- One Trigger per Object
- Logic should be in Handler class, not in Trigger
- Always bulkify your code
- Avoid SOQL and DML inside loops

Q37: What is a Trigger Handler? Ans: Trigger Handler is a separate Apex class that contains the actual business logic. The trigger just calls methods from this class. This keeps code clean and organized.

Q38: What is recursion in Triggers? Ans: Recursion happens when a trigger calls itself repeatedly in a loop. Example: Update trigger updates a record, which fires the same trigger again. This can cause errors.

Q39: How to prevent Trigger recursion? Ans: Use a static Boolean variable in a helper class to track if trigger has already run. Check this variable at the beginning of your trigger.

SOQL (Salesforce Object Query Language)

Q40: What is SOQL? Ans: SOQL is Salesforce Object Query Language. It's used to query/retrieve records from Salesforce database. Similar to SQL SELECT statement.

Q41: Write a simple SOQL query? Ans:

SELECT Name, Email FROM Contact WHERE AccountId = '001xxxxxx'

Q42: What is the difference between SOQL and SQL? Ans: SOQL only supports SELECT queries (no INSERT, UPDATE, DELETE). SOQL queries one object at a time. SOQL can query related objects using relationships.

Q43: What are SOQL keywords? Ans: SELECT, FROM, WHERE, ORDER BY, LIMIT, OFFSET, GROUP BY, HAVING, LIKE, IN, NOT IN, AND, OR

Q44: What is the maximum number of records SOQL can return? Ans: 50,000 records in one query.

Q45: How to query related records in SOQL? Ans:

- Parent to Child: `SELECT Name, (SELECT LastName FROM Contacts) FROM Account`
- Child to Parent: `SELECT LastName, Account.Name FROM Contact`

Q46: What is SOQL injection? Ans: SOQL injection is a security vulnerability where malicious code is inserted into SOQL queries. Always use bind variables (:variable) to prevent it.

Q47: What are SOQL best practices? Ans:

- Use WHERE clause to filter records
- Use LIMIT to reduce records
- Never write SOQL inside loops
- Use bind variables to prevent injection

- Query only required fields

SOSL (Salesforce Object Search Language)

Q48: What is SOSL? Ans: SOSL is Salesforce Object Search Language. It's used to search text across multiple objects at once. Good for search boxes.

Q49: What is the difference between SOQL and SOSL? Ans:

- SOQL: Query one object, returns records, more specific
- SOSL: Search multiple objects, returns list of lists, good for text search

Q50: Write a simple SOSL query? Ans:

FIND {John} IN ALL FIELDS RETURNING Account(Name), Contact(FirstName, LastName)

DML (Data Manipulation Language)

Q51: What is DML? Ans: DML stands for Data Manipulation Language. It's used to insert, update, delete, and undelete records in Salesforce database.

Q52: What are DML operations? Ans:

- insert - create new records
- update - modify existing records
- delete - delete records
- undelete - restore deleted records
- upsert - update if exists, insert if not
- merge - merge duplicate records

Q53: What is the difference between Database methods and DML statements? Ans:

- DML statements (insert, update): If one record fails, entire transaction fails
- Database methods (Database.insert, Database.update): Can allow partial success using allOrNone parameter

Q54: What is the maximum DML limit? Ans: 150 DML statements per transaction.

Q55: What are DML best practices? Ans:

- Never use DML inside loops
- Bulkify your code (process multiple records together)
- Use try-catch for error handling
- Check record limits before DML

GOVERNOR LIMITS

Q56: What are Governor Limits? Ans: Governor Limits are Salesforce's way of ensuring no single code hogs shared resources. They limit how much data and operations your code can use.

Q57: What are important Governor Limits? Ans:

- Total SOQL queries: 100
- Total records retrieved by SOQL: 50,000
- Total DML statements: 150
- Total records processed by DML: 10,000
- Total Heap size: 6 MB (synchronous), 12 MB (asynchronous)
- Maximum CPU time: 10 seconds (synchronous), 60 seconds (asynchronous)

Q58: How to avoid Governor Limits? Ans:

- Bulkify your code
- Never put SOQL or DML inside loops
- Use collections to process multiple records
- Use asynchronous Apex for large data
- Use efficient queries with WHERE clause

Q59: What is Bulkification? Ans: Bulkification means writing code that can handle multiple records at once efficiently. Instead of processing one record at a time, process all records together.

ASYNCHRONOUS APEX

Q60: What is Asynchronous Apex? Ans: Asynchronous Apex runs in the background, separate from the main transaction. It has higher limits and doesn't make users wait.

Q61: What are types of Asynchronous Apex? Ans:

- Future Methods
- Batch Apex
- Queueable Apex
- Scheduled Apex

Q62: What are Future Methods? Ans: Future Methods run in a separate thread in the background. They're annotated with @future. Used for callouts to external systems and long-running processes.

Q63: What are limitations of Future Methods? Ans:

- Cannot call another future method
- Cannot track execution status
- Parameters must be primitive types (no sObjects)
- Limited to 50 method calls per transaction

Q64: What is Batch Apex? Ans: Batch Apex processes large numbers of records in chunks. It implements Database.Batchable interface. Has three methods: start, execute, finish.

Q65: When to use Batch Apex? Ans:

- Processing millions of records
- Data cleanup jobs
- Mass updates
- Complex calculations on large datasets

Q66: What is Queueable Apex? Ans: Queueable Apex is similar to future methods but better. You can chain jobs, pass complex types, and track job status. Implements Queueable interface.

Q67: What is Scheduled Apex? Ans: Scheduled Apex runs at specific times (like a cron job). It implements Schedulable interface. You can schedule it to run daily, weekly, or at specific times.

VISUALFORCE

Q68: What is Visualforce? Ans: Visualforce is a framework for building custom user interfaces in Salesforce. It uses tag-based markup similar to HTML.

Q69: What is a Visualforce Page? Ans: A Visualforce Page is a custom page created using Visualforce tags. It displays data and provides custom UI.

Q70: What is a Visualforce Controller? Ans: Controller is the Apex class that contains the logic and data for a Visualforce page. It handles button clicks, data retrieval, and business logic.

Q71: What are types of Controllers? Ans:

- Standard Controller - automatically provided for standard/custom objects
- Custom Controller - Apex class you create
- Controller Extension - extends standard controller with custom logic

Q72: What is the difference between Standard and Custom Controller? Ans:

- Standard Controller: Automatic, no code needed, basic operations only
- Custom Controller: Complete control, custom logic, must write all code yourself

Q73: What are important Visualforce tags? Ans:

- `<apex:page>` - defines the page
- `<apex:form>` - creates a form
- `<apex:pageBlock>` - section like standard Salesforce pages
- `<apex:inputField>` - input field
- `<apex:outputField>` - display field
- `<apex:commandButton>` - button
- `<apex:pageMessages>` - display error messages

LIGHTNING WEB COMPONENTS (LWC)

Q74: What is Lightning Web Component? Ans: LWC is a modern framework for building components in Salesforce using standard web technologies (HTML, JavaScript, CSS). It's faster and easier than Aura.

Q75: What are the files in an LWC? Ans:

- HTML file - structure/template
- JavaScript file - logic
- CSS file - styling (optional)
- XML file - metadata configuration

Q76: What is the difference between Aura and LWC? Ans:

- LWC: Uses standard JavaScript, faster, easier to learn, modern
- Aura: Salesforce-specific, older, more complex

Q77: What are decorators in LWC? Ans:

- @api - makes property/method public
- @track - makes property reactive (auto-updates UI)
- @wire - calls Apex methods or gets data

Q78: What is @wire decorator? Ans: @wire is used to read Salesforce data automatically. It calls Apex methods or uses Lightning Data Service to get records.

Q79: What is the lifecycle of LWC? Ans:

- constructor() - component is created
- connectedCallback() - component inserted into DOM
- renderedCallback() - after render
- disconnectedCallback() - component removed from DOM

SALES CLOUD SPECIFIC

Q80: What is a Lead? Ans: Lead is a potential customer who has shown interest but hasn't been qualified yet. Contains basic information like name, company, email.

Q81: What is Lead Conversion? Ans: Lead Conversion is the process of converting a qualified lead into Account, Contact, and optionally an Opportunity.

Q82: What is an Opportunity? Ans: Opportunity represents a potential sale or deal. It tracks the sales process from start to close with stages and amount.

Q83: What are Opportunity Stages? Ans: Stages track progress of a deal. Examples: Prospecting, Qualification, Proposal, Negotiation, Closed Won, Closed Lost.

Q84: What is a Quote? Ans: Quote is a formal price proposal you send to customers. It lists products, quantities, and prices.

Q85: What is Territory Management? Ans: Territory Management allows companies to organize accounts and opportunities based on geographical regions or other criteria.

Q86: What is a Campaign? Ans: Campaign is a marketing initiative to generate leads or increase brand awareness. You can track responses and ROI.

Q87: What is Forecast? Ans: Forecast allows sales managers to predict future sales revenue based on opportunities in the pipeline.

Q88: What are Products and Price Books? Ans: Products are items/services you sell. Price Books contain prices for products. You can have multiple price books for different regions or customer types.

SERVICE CLOUD SPECIFIC

Q89: What is a Case? Ans: Case represents a customer question, problem, or feedback. It's the main object in Service Cloud for tracking customer issues.

Q90: What is Case Assignment Rule? Ans: Case Assignment Rule automatically assigns cases to users or queues based on criteria like case type, priority, or source.

Q91: What is Escalation Rule? Ans: Escalation Rule automatically escalates cases that aren't resolved within a certain time. It can notify managers or change priority.

Q92: What is a Queue? Ans: Queue is a holding area for records that need to be worked on. Multiple users can access and pick cases from a queue.

Q93: What is Email-to-Case? Ans: Email-to-Case automatically creates cases from customer emails sent to a specific email address. It helps track all customer communications.

Q94: What is Web-to-Case? Ans: Web-to-Case creates cases from website forms. Customers submit issues through your website and cases are automatically created.

Q95: What is Live Agent? Ans: Live Agent is a live chat feature that allows customers to chat in real-time with support agents on your website.

Q96: What is Omni-Channel? Ans: Omni-Channel routes different types of work items (cases, chats, leads) to available agents based on their skills and capacity.

Q97: What is Knowledge Base? Ans: Knowledge Base is a collection of articles that provide solutions to common problems. Agents and customers can search for answers.

Q98: What is Entitlement? Ans: Entitlement defines the level of support a customer receives based on their contract. It specifies things like response time and available support channels.

Q99: What is Service Level Agreement (SLA)? Ans: SLA is a commitment to customers about response and resolution times for cases. Milestones track if SLA is met.

Q100: What is a Milestone? Ans: Milestone is a required step in the case resolution process with a deadline. Example: "First Response" within 2 hours.

AUTOMATION TOOLS

Q101: What is Workflow Rule? Ans: Workflow Rule is an automation tool that triggers actions (email alerts, field updates, tasks) when certain conditions are met. Now being replaced by Flow.

Q102: What is Process Builder? Ans: Process Builder is a point-and-click tool for creating automated processes. More powerful than workflows. Now being replaced by Flow.

Q103: What is Flow? Ans: Flow is the most powerful automation tool in Salesforce. It can do everything workflows and process builder can do, plus much more. Uses a visual designer.

Q104: What are types of Flows? Ans:

- Screen Flow - with user interface
- Autolaunched Flow - runs in background
- Record-Triggered Flow - runs when record is created/updated
- Scheduled Flow - runs at specific times
- Platform Event-Triggered Flow

Q105: What is the difference between Workflow, Process Builder, and Flow? Ans:

- Workflow: Simple, limited actions
- Process Builder: More powerful, can update related records

- Flow: Most powerful, can do complex logic, loops, screens

INTEGRATION

Q106: What is API? Ans: API (Application Programming Interface) allows external systems to connect and communicate with Salesforce to exchange data.

Q107: What are types of Salesforce APIs? Ans:

- REST API - modern, uses HTTP methods, lightweight
- SOAP API - older, more complex, very secure
- Bulk API - for large data operations
- Streaming API - for real-time events
- Metadata API - for deployments

Q108: What is Callout? Ans: Callout is when Salesforce code calls an external web service. Used to get data from or send data to external systems.

Q109: What is Remote Site Setting? Ans: Remote Site Setting is a security feature. You must add external URLs here before your Apex code can call them.

Q110: What is Named Credential? Ans: Named Credential stores the URL and authentication details for external systems. It makes callouts easier and more secure.

TESTING

Q111: What is a Test Class? Ans: Test Class is an Apex class that tests your code. It's annotated with @isTest. You must write test classes to deploy code to production.

Q112: What is code coverage? Ans: Code Coverage is the percentage of your code that is tested. You need at least 75% code coverage to deploy to production.

Q113: What are test methods? Ans: Test methods are methods inside test class that test your code. They're annotated with @isTest or testMethod keyword.

Q114: What is Test.startTest() and Test.stopTest()? Ans: These methods reset governor limits for the code between them. Useful for testing triggers and batch jobs. Code after startTest gets fresh limits.

Q115: What is System.assert()? Ans: System.assert() verifies that a condition is true. If false, test fails. Used to check if code produces expected results.

Q116: What is @testSetup? Ans: @testSetup method creates test data once that all test methods in the class can use. It saves resources and makes tests faster.

Q117: What is Test.isRunningTest()? Ans: This returns true if code is running in test context. Used to skip certain logic during testing.

DATA MANAGEMENT

Q118: What is Data Loader? Ans: Data Loader is a desktop application for bulk import/export of data. It can handle up to 5 million records.

Q119: What is Import Wizard? Ans: Import Wizard is a simple tool in Salesforce setup for importing data. Limited to 50,000 records. Good for beginners.

Q120: What is the difference between Data Loader and Import Wizard? Ans:

- Data Loader: 5 million records, command-line support, all objects
- Import Wizard: 50,000 records, web-based, limited objects

Q121: What is External ID? Ans: External ID is a custom field marked as unique identifier. Used for integrations and upsert operations to match records.

Q122: What is Upsert? Ans: Upsert operation updates existing records or inserts new ones. It uses External ID to determine if record exists.

MISCELLANEOUS IMPORTANT TOPICS

Q123: What is Sandbox? Ans: Sandbox is a copy of your production environment used for testing and development. Changes here don't affect real data.

Q124: What are types of Sandboxes? Ans:

- Developer Sandbox - small, for coding
- Developer Pro Sandbox - larger storage
- Partial Copy Sandbox - includes sample data
- Full Sandbox - complete copy of production

Q125: What is Change Set? Ans: Change Set is used to move customizations from one Salesforce org to another. Used for deploying code without third-party tools.

Q126: What is Static Resource? Ans: Static Resource stores files (images, JavaScript, CSS) that you can use in Visualforce or Lightning components.

Q127: What is Validation Rule? Ans: Validation Rule ensures data quality by checking if data meets certain conditions before saving. Example: Email must contain "@".

Q128: What is Formula Field? Ans: Formula Field is a read-only field that automatically calculates its value using other fields. Example: Full Name = First Name + Last Name.

Q129: What is Custom Metadata Type? Ans: Custom Metadata Type is like custom object but it stores configuration data that can be deployed. It's similar to custom settings but better.

Q130: What is Custom Setting? Ans: Custom Setting stores custom data accessible across your organization. Two types: List (multiple records) and Hierarchy (user-specific).

Q131: What is Page Layout? Ans: Page Layout controls which fields, related lists, and buttons appear on a record page for different profiles.

Q132: What is Record Id? Ans: Record Id is a unique 15 or 18-character identifier for each record in Salesforce. 15-char is case-sensitive, 18-char is case-insensitive.

Q133: What is Email Template? Ans: Email Template is a pre-designed email format you can use to send emails from Salesforce. Can include merge fields.

Q134: What is Dynamic SOQL? Ans: Dynamic SOQL builds query as a string at runtime. Useful when query conditions change. Created using Database.query() method.

Q135: What is Lightning App Builder? Ans: Lightning App Builder is a drag-and-drop tool to build custom Lightning pages by adding components.

Q136: What is Lightning Record Page? Ans: Lightning Record Page is a custom layout for viewing and editing records using Lightning components.

Q137: What is Approval Process? Ans: Approval Process automates record approvals. Records are submitted, routed to approvers, and then approved or rejected.

Q138: What is Big Objects? Ans: Big Objects store massive amounts of data (billions of records) for archiving and analytics. Accessed only through SOQL.

Q139: What is Platform Events? Ans: Platform Events are used for event-driven architecture. They allow different parts of Salesforce or external systems to communicate.

Q140: What is Limits class? Ans: Limits class provides methods to check current governor limit usage. Example: Limits.getQueries() returns number of SOQL queries used.

SCENARIO-BASED QUESTIONS

Q141: How would you send an email when an Opportunity is closed won? Ans: Create a Record-Triggered Flow on Opportunity. Set trigger to "when record is created or updated". Add condition: Stage = "Closed Won". Add action: Send Email.

Q142: How would you prevent duplicate Contacts based on email? Ans:

1. Create a Validation Rule checking if email already exists
2. Or make Email field unique in field settings
3. Or use duplicate rules

Q143: How would you update all Accounts at once? Ans:

1. For small data: Use Data Loader or Flow
2. For large data: Use Batch Apex

Q144: How would you create a field that shows Account Name on Contact? Ans: Create a Formula Field on Contact with formula: Account.Name

Q145: How would you show total Opportunity Amount on Account? Ans: Create a Roll-Up Summary field on Account that sums Amount from Opportunities.

Q146: A trigger is firing multiple times. How to fix it? Ans: Create a static Boolean variable in a helper class. Check if trigger already ran. If yes, return immediately.

Q147: How to call an external API from Salesforce? Ans:

1. Add URL to Remote Site Settings
2. Write Apex code with Http callout
3. Use @future method if calling from trigger

Q148: User can't see a button. Why? Ans:

1. Check if button is on page layout
2. Check profile permissions
3. Check if button has visibility criteria

Q149: How to show data from multiple objects on one page? Ans:

1. Use Visualforce with custom controller
2. Use Lightning Web Component
3. Use Related Lists on Lightning page

Q150: How would you schedule a job to run every night? Ans: Write a class implementing Schedulable interface. Schedule it from Setup → Scheduled Jobs or use System.schedule() in code.

BEST PRACTICES

Q151: What are Apex coding best practices? Ans:

- Bulkify your code
- Avoid SOQL/DML in loops
- Use proper error handling
- Follow naming conventions
- Write test classes
- Comment your code
- Use trigger framework

- Check governor limits

Q152: What are trigger best practices? Ans:

- One trigger per object
- Use trigger handler class
- Bulkify code
- Avoid recursion
- Handle all events in one trigger

Q153: What are security best practices? Ans:

- Use CRUD and FLS checks
- Prevent SOQL injection
- Use with sharing keyword
- Encrypt sensitive data
- Use field-level security

Q154: What are performance best practices? Ans:

- Use selective queries
- Index fields used in WHERE
- Avoid query all rows
- Use pagination for large data
- Cache reusable data
