---

1. What are different types of relationships in Salesforce?
Answer:

- **Lookup Relationship**: Loose relationship between objects; child can exist without parent.
- **Master-Detail Relationship**: Strong dependency; child record is deleted if parent is deleted.
- **Self-Relationship**: Object related to itself.
- **Many-to-Many**: Implemented using a **junction object** with two master-detail relationships.
- **External Lookup / Indirect Lookup**: Used with external objects for integrations.

---

2. What is the difference between Role Hierarchy, Sharing Rules, and Profiles?
Answer:

- **Profiles** → Define object-level and field-level access (CRUD + FLS).
- **Role Hierarchy** → Provides record-level access upward in hierarchy.
- **Sharing Rules** → Open up access horizontally across roles.
- **Permission Sets** → Extend permissions beyond profiles without changing the profile.

---

3. Difference between Workflow, Process Builder, and Flow?
Answer:

- **Workflow** → Limited automation (field updates, email alerts, tasks).
- **Process Builder** → More powerful, can update related records, create records, invoke flows.
- **Flow (Lightning Flow)** → Most powerful; can handle complex business logic, loops, DML, screen interactions.
  - *Note:* Salesforce is retiring Workflow and Process Builder → Flows are the future.

---

4. Can we make a field required in three different ways?
Answer:
Yes:

1. **Field level settings (while creating field)** → universally required.
2. **Page layout** → required only on UI.
3. **Validation Rule** → conditional requirement.

---

5. What are Governor Limits in Salesforce?
Answer:
Governor limits are runtime restrictions enforced by Salesforce to ensure multi-tenant efficiency.
Examples:

- SOQL queries: **100 per transaction** (synchronous)
- DML statements: **150 per transaction**
- CPU time: **10,000 ms**
- Heap size: **6 MB (sync), 12 MB (async)**
- Records retrieved by SOQL: **50,000**

---

6. What is a Trigger? Best Practices?

Answer:

- Trigger is Apex code that executes **before or after** DML events (insert, update, delete, undelete).
- **Best practices**:
  - Use **context variables** (Trigger.new, Trigger.old).
  - Avoid **SOQL/DML inside loops**.
  - Use **Handler classes** for logic (Trigger Framework).
  - Ensure **bulkification** (handle multiple records).
  - Add **recursion handling** if needed.

---

7. Can you explain Trigger.new vs Trigger.old?
Answer:

- **Trigger.new** → Holds new versions of records (available in insert, update).
- **Trigger.old** → Holds old versions of records (available in update, delete).
- **Example:** In update trigger, compare Trigger.oldMap.get(Id) with Trigger.newMap.get(Id) to detect changes.

---

8. Difference between SOQL and SOSL?
Answer:

- **SOQL** → Query records from **a single object or related objects**. (Example: SELECT Id, Name FROM Account WHERE Name = 'ABC').
- **SOSL** → Text search across **multiple objects** and fields simultaneously. Returns limited fields (Id, Name).

---

9. Explain LWC vs Aura? Why LWC?
Answer:

- **Aura Components** → Older framework, XML + JS, event-driven.
- **LWC (Lightning Web Components)** → Built on modern JS standards (ES6, Shadow DOM, Web Components).
- **Why LWC?**
  - Faster performance.
  - Lightweight.
  - Uses native browser APIs.
  - Better testability & reusability.

---

10. How do you handle large data volume in Salesforce?
Answer:

- Use **Batch Apex** (process records in batches of 200).
- Use **Queueable Apex** for async operations.
- Use **Future methods** for lightweight async tasks.
- Use **Indexes** on filter fields to improve SOQL.
- Use **Skinny tables** for performance.
- Apply **Selective Queries** (avoid !=, avoid null checks without index).

---

11. Difference between API types: REST vs SOAP in Salesforce?
Answer:

- **REST API** → Lightweight, JSON/XML, stateless, best for mobile/web apps.
- **SOAP API** → XML-based, strict schema, enterprise integrations (ERP/Finance systems).
- **Use case**: REST for modern apps, SOAP for legacy enterprise integrations.

---

12. What are different types of sandboxes?
Answer:

- **Developer Sandbox** → Small, limited data.
- **Developer Pro** → Larger storage.
- **Partial Copy** → Metadata + sample data (templates).
- **Full Copy** → Complete production replica with all data.

---

13. What is Deployment in Salesforce? What tools have you used?
Answer:

- Deployment = Moving metadata/code from **Sandbox → Production**.
- Tools:
  - **Change Sets** (Admin-friendly, UI-based).
  - **ANT Migration Tool** (scripted, automated).
  - **Salesforce CLI (SFDX)** (modern, DevOps-friendly).
  - **Copado/Gearset/Jenkins** (CI/CD tools).

---

14. How do you enforce field-level security in Apex?
Answer:

- **Schema methods**: Schema.sObjectType.Account.fields.Name.isAccessible()
- Use **with sharing** keyword in Apex classes.
- Apply **CRUD/FLS checks** before performing DML or queries.

---

15. Real-Time Scenario Question:
Suppose when an Opportunity is Closed-Won, you need to create a Contract automatically. How will you implement this?
Answer:

- Use **After Insert/Update Trigger** on Opportunity.
- OR use **Record-Triggered Flow** (preferred low-code approach).
- Logic: Check Opportunity.StageName == 'Closed Won', then create Contract record with related fields.

---

16. How do you avoid recursive triggers?
Answer:

- Use a **static Boolean variable** in a helper class.
- Before performing DML, check if the flag is already set.
- Example:

```
public class TriggerHelper {
   public static Boolean isFirstRun = true;
}
trigger AccountTrigger on Account (after update) {
   if(TriggerHelper.isFirstRun) {
      TriggerHelper.isFirstRun = false;
```

```
        // logic here
    }
}
```

---

17. Explain Difference between Standard Controller, Custom Controller, and Controller Extension in Salesforce.

Answer:

- **Standard Controller** → Provides out-of-box logic for a standard/ custom object.
- **Custom Controller** → Fully customized Apex class controlling a VF page.
- **Controller Extension** → Extends functionality of a standard/custom controller without rewriting it.

---

18. What is a Wrapper Class? Use Case?

Answer:
A **wrapper class** is a custom class that combines multiple data types into a single object.

- **Use case:**
    - Display checkbox with each record in a VF/LWC table.
    - Combine Account + related Contacts into one structure.

---

19. How do you call Apex from LWC?

Answer:

- Import method from Apex class with @AuraEnabled annotation.
- Example:

```
import getAccounts from '@salesforce/apex/AccountController.getAccounts';
```

- Call inside JS file:

```
getAccounts()
.then(result => { this.accounts = result; })
.catch(error => { this.error = error; });
```

---

20. How do you manage errors and exceptions in Apex?

Answer:

- Use **try-catch-finally**.
- Catch **DmlException, NullPointerException, QueryException**.
- Use **addError()** method to show error on UI.
- Store errors in **Custom Object / Log** for debugging.

21. What are Record Types and why do we use them?

Answer:

- Record Types allow multiple **business processes** on the same object.
- Control **picklist values**, **page layouts**, and **record-level business logic**.
- Example: On Opportunity → different sales processes for *Enterprise vs SMB* deals.

---

22. Difference between Public Group and Queue?

Answer:

- **Public Group**: Collection of users, roles, and other groups → used for sharing rules.
- **Queue**: Holds records until assigned → mainly used for *Lead, Case, Custom Object* assignments.

---

23. What is OWD?

Answer:
Organization-Wide Defaults (OWD) → baseline record-level access. Options:

- **Private** → Only owner + above in role hierarchy.
- **Public Read Only** → Everyone can see but not edit.
- **Public Read/Write** → Everyone can see and edit.
- **Controlled by Parent** → Access inherits from parent object.

---

24. Difference between Standard Profile and Custom Profile?

Answer:

- **Standard Profiles** (like System Administrator, Standard User) → can't delete or rename. Limited modification.
- **Custom Profiles** → fully customizable (object/field permissions, login hours, etc.).

---

25. What is the difference between Permission Set and Permission Set Group?

Answer:

- **Permission Set** → Add permissions to users without changing profile.
- **Permission Set Group** → Bundle multiple permission sets into one assignable group. Useful for large orgs.

---

26. What is the difference between Data Loader and Data Import Wizard?

Answer:

- **Data Import Wizard**:
  - Web-based UI.
  - Supports limited objects.
  - Up to 50,000 records.
- **Data Loader**:
  - Client application.
  - Supports all objects.
  - Up to 5 million records.
  - Supports automation (command line).

---

27. How do you ensure data quality in Salesforce?

Answer:

- **Validation Rules**
- **Duplicate Rules & Matching Rules**
- **Required fields**
- **Reports & Dashboards** for monitoring
- **Third-party tools (DemandTools, Informatica)**

---

28. What are Dynamic Forms in Salesforce?

Answer:

- Feature in Lightning → allows admins to place fields & sections directly on Lightning Record Pages (not just layouts).
- Provides **conditional visibility** & **page-level flexibility**.

---

29. What are Salesforce Shield features?

Answer:

- **Field Audit Trail** → Track up to 10 years of field history.
- **Shield Platform Encryption** → Data encryption at rest.
- **Event Monitoring** → Track user activity & API usage.

---

30. What are Assignment Rules in Salesforce?

Answer:

- Rules that automatically assign **Leads** or **Cases** to users or queues based on conditions.
- Can define multiple rules but only one active rule at a time.

APEX & DEVELOPMENT QUESTIONS

31. What is the difference between "with sharing" and "without sharing"?

Answer:

- **with sharing** → Enforces current user's sharing rules.
- **without sharing** → Runs in system context, bypassing sharing.
- **Default**: Apex classes run in system mode unless specified.

---

32. What are Future Methods in Apex?

Answer:

- Annotated with @future → run asynchronously.
- Use cases: Callouts, async DML, mass updates.
- Limitations: Must be static, return void, max 50 per transaction.

---

33. Difference between Batch Apex and Queueable Apex?

Answer:

- **Batch Apex** → For large data (millions of records), runs in batches of 200.
- **Queueable Apex** → Lightweight async jobs, allows chaining, supports complex objects.

---

34. Explain Apex Transactions.

Answer:

- A single unit of work in Salesforce → includes all DML, SOQL, logic.
- Either **commits** (if no error) or **rolls back** (if exception).
- Governor limits apply per transaction.

---

35. What is Apex Test Class?

Answer:

- Apex tests ensure code quality.
- Must cover at least **75%** of Apex code before deployment to Production.
- Best practices: Test bulk, test positive/negative, use System.assert(), avoid hard-coded Ids.

---

36. Explain Difference between Before Trigger and After Trigger.

Answer:

- **Before Trigger** → Used to update/validate values before saving (e.g., auto populate fields).
- **After Trigger** → Used when record Id is needed (e.g., create related records).

---

37. What are Custom Metadata Types vs Custom Settings?

Answer:

- **Custom Settings** → Store configuration data (accessible in Apex without SOQL). Two types: List & Hierarchy.
- **Custom Metadata Types** → Metadata records deployable across orgs, better for configuration across environments.

---

38. What is Invocable Method in Apex?

Answer:

- Annotated with @InvocableMethod.
- Exposed to **Flows** and **Process Builder**.
- Allows low-code admins to call Apex logic.

---

39. What is the difference between Database.insert() and insert DML?

Answer:

- **insert** → Throws error if any record fails.
- **Database.insert(list, false)** → Partial success allowed, returns SaveResult[].

---

40. What is an Apex Trigger Framework?

Answer:

- A structured way to write triggers:
  - One trigger per object.
  - Delegate logic to handler classes.
  - Handle multiple events.
  - Avoid recursion.
- Example: **Handler pattern**.

LWC (Lightning Web Components)

41. What is Reactive Property in LWC?

Answer:

- Properties decorated with @track, @api, or @wire.
- UI automatically updates when values change.

---

42. What is the difference between @api, @track, and @wire in LWC?

Answer:

- **@api** → Expose property/method to parent component.
- **@track** → Track private reactive state (default in modern LWC).
- **@wire** → Call Salesforce data/services declaratively (Apex, LDS, UI API).

---

43. How do you perform Apex callouts in LWC?

Answer:

- Cannot call external API directly.
- Must create **Apex method with @AuraEnabled** → inside that use HttpRequest/HttpResponse.
- Call from LWC JS using imported Apex method.

---

44. What is LDS (Lightning Data Service)?

Answer:

- Provides record CRUD without Apex/SOQL.
- Example: lightning-record-form, lightning-record-edit-form.
- Handles FLS, sharing rules automatically.

---

45. How do you handle navigation in LWC?

Answer:

- Use NavigationMixin.
- Example: Navigate to record page:

```
this[NavigationMixin.Navigate]({
    type: 'standard__recordPage',
    attributes: {
        recordId: this.recordId,
        objectApiName: 'Account',
        actionName: 'view'
    }
});
```

INTEGRATION & DEPLOYMENT

46. What is Outbound Message in Salesforce?

Answer:

- Workflow action that sends SOAP message to external system.
- Used in simple point-to-point integration.
- No code required.

---

47. Explain Named Credentials.

Answer:

- Store **endpoint URL + authentication** in Salesforce securely.
- Simplifies integration without storing passwords in code.

---

48. What is Change Data Capture (CDC)?

Answer:

- Real-time event streaming for record changes.
- Sends insert/update/delete/undelete events to external systems.
- Useful for integrations (Kafka, MuleSoft).

---

49. How do you move metadata between environments?

Answer:

- **Change Sets**
- **ANT Migration Tool**
- **Salesforce CLI (SFDX)**
- **CI/CD tools like Copado, Gearset, Jenkins**

---

50. Explain difference between Managed and Unmanaged Packages.

Answer:

- **Managed** → Published on AppExchange, versioned, protected IP.
- **Unmanaged** → Source code visible, used for sharing/reuse, no version upgrades.

REAL-TIME SCENARIOS

51. Scenario:

If user updates Account Billing Address, automatically update all related Contact addresses.

Answer:

- **Best option today**: Record-Triggered Flow on Account → Loop contacts → Update fields.
- Alternative: After Update Trigger on Account.

---

52. Scenario:

When a Case is closed, send email to customer and create a follow-up Task for agent.

Answer:

- Use **Record-Triggered Flow** (preferred).
- Entry condition: Case.Status = Closed.
- Actions: Send Email → Create Task.

---

53. Scenario:

Prevent Opportunity from moving to "Closed Won" if "Amount" < 5000.

Answer:

- **Validation Rule**:

AND(ISPICKVAL(StageName,'Closed Won'), Amount < 5000)

---

54. Scenario:

Auto-assign high priority Cases to "Priority Support Queue".

Answer:

- Create **Case Assignment Rule** → Condition: Priority = High → Queue = Priority Support.

---

55. Scenario:

If user deletes Account, prevent deletion if related Opportunities exist.

Answer:

- Write **Before Delete Trigger** on Account.
- Check if SELECT Id FROM Opportunity WHERE AccountId IN :Trigger.oldMap.keySet().
- If exists → addError().

56. What is the difference between Database.executeBatch and System.scheduleBatch?

Answer:

- **Database.executeBatch** → Run batch job immediately.
- **System.scheduleBatch** → Schedule a batch job at a specific time (cron expression).

---

57. What are Custom Labels in Salesforce?

Answer:

- Store **text values** (like error messages, UI text, API endpoints).
- Supports **translations**.
- Accessible in Apex using: System.Label.LabelName

---

58. How do you handle callouts from Batch Apex?

Answer:

- Batch class must implement **Database.AllowsCallouts** interface.
- Allows HTTP callouts within each batch execution.

---

59. Explain difference between Synchronous and Asynchronous Apex.

Answer:

- **Synchronous Apex** → Executes immediately (Triggers, Classes, VF Controllers).
- **Asynchronous Apex** → Executes in background (Future methods, Batch Apex, Queueable, Schedulable).

---

60. What is the difference between Queueable and Future methods?

Answer:

- Queueable supports:
  - Chaining jobs.
  - Complex objects as parameters.
- Future method:
  - Lightweight, cannot be chained.
  - Only primitive types allowed.

---

61. What are Platform Events?

Answer:

- Event-driven architecture in Salesforce.
- Publishers fire events, subscribers consume.
- Used for **real-time integrations** and async processes.

---

62. What is the difference between Standard and Custom Metadata Records deployment?

Answer:

- **Standard Metadata** → Like objects, fields, workflows (moved via Change Sets/ANT/SFDX).
- **Custom Metadata Records** → Deployable like config data (unlike Custom Settings which are data, not metadata).

---

63. How do you handle recursion in Flows?

Answer:

- Use **Flow entry criteria** carefully.
- Example: Instead of "on every update," check if a specific field changed → ISCHANGED() condition.
- Or add **Decision element** to exit when not needed.

SECURITY & SHARING

64. What is the difference between Sharing Rule and Manual Sharing?

Answer:

- **Sharing Rule** → Automates record-level sharing (criteria/owner-based).
- **Manual Sharing** → One-off, user manually shares record with others.

---

65. How do you enforce CRUD/FLS in Apex?

Answer:

- Check permissions before DML or SOQL. Example:

```
if(Schema.sObjectType.Account.isCreateable()){
   insert acc;
}
```

- For fields:

```
if(Schema.sObjectType.Account.fields.Name.isAccessible()){
   System.debug(acc.Name);
}
```

---

66. Difference between Profile Login Hours vs IP Restrictions?

Answer:

- **Login Hours** → Restrict login time. Outside allowed time, user cannot log in.
- **IP Restrictions** → Restrict by IP range. Outside range, login blocked.

---

67. What is the difference between “with sharing” and “inherited sharing”?

Answer:

- **with sharing** → Always enforce sharing.
- **inherited sharing** → Enforces sharing depending on caller context (flexible, recommended for reusable classes).

---

68. Explain Organization-Wide Defaults (OWD) vs Permission Sets.

Answer:

- **OWD** → Defines baseline access for everyone.
- **Permission Sets** → Give additional access to specific users without changing profile.

DATA MANAGEMENT

69. What is Skinny Table in Salesforce?

Answer:

- A custom-built table maintained by Salesforce to improve query performance.
- Contains frequently used fields from a standard/custom object.
- Read-only to admins.

---

70. Difference between Big Objects and Custom Objects?

Answer:

- **Custom Object** → Normal records (CRUD, reports, triggers).
- **Big Object** → Handle **billions** of records, optimized for storage + read, limited fields & operations, async queries.

---

71. How do you prevent data duplication?

Answer:

- **Duplicate Rules** + **Matching Rules**.
- **Validation Rules**.
- **Custom Trigger**.
- **Third-party tools** (DemandTools, Informatica).

---

72. What are External Objects in Salesforce?

Answer:

- Represent data stored outside Salesforce (via OData).
- Access external DBs without storing data inside Salesforce.
- Example: Connect SAP/Oracle tables directly.

INTEGRATIONS

73. What is Named Credential vs Remote Site Settings?

Answer:

- **Remote Site Setting** → Whitelist external endpoint (mandatory before callouts).
- **Named Credential** → Stores endpoint + authentication securely, easier to manage.

---

74. What is Salesforce Connect?

Answer:

- Integration tool to access external data in **real time** via External Objects.
- No data duplication → data stays in external system.

---

75. How do you handle authentication for external integrations?

Answer:

- Options: **OAuth 2.0, Named Credentials, JWT, Password-based, Session-based tokens.**

---

76. Difference between Platform Events and Change Data Capture (CDC)?

Answer:

- **Platform Events** → Custom events for event-driven apps.
- **CDC** → System events for record-level changes (insert/update/delete/undelete).

---

77. What is an Outbound Message limitation?

Answer:

- SOAP only (no REST).
- Fire-and-forget (no response handling).
- Retry logic limited.

DEPLOYMENT & DEVOPS

78. What is the difference between ANT and Salesforce CLI (SFDX)?

Answer:

- **ANT** → Older Java-based tool, script-heavy.
- **SFDX CLI** → Modern, supports source-driven development, scratch orgs, CI/CD.

---

79. Explain Scratch Orgs.

Answer:

- Disposable Salesforce environments.
- Used in SFDX for **development/testing**.
- Easily created & destroyed via CLI.

---

80. What is the difference between Change Set vs Unlocked Package?

Answer:

- **Change Set** → Click-based, only within related orgs.

- **Unlocked Package** → Source-driven, modular, reusable, works across orgs, versioned.

REAL-TIME SCENARIOS

81. Scenario:

When a new Contact is created with "VIP = True", auto-create a Case assigned to "VIP Support Queue".

Answer:

- **Record-Triggered Flow** on Contact → Condition VIP__c = True → Create Case → Assign to Queue.

---

82. Scenario:

If Opportunity is Closed Won, update parent Account field "Last_Deal_Date__c" with Close Date.

Answer:

- **After Update Flow** on Opportunity.
- OR **After Update Trigger** if complex.

---

83. Scenario:

Block users from deleting a Case if Status = "Escalated".

Answer:

- **Before Delete Trigger** → addError() if Status == 'Escalated'.

---

84. Scenario:

Send Slack notification when a High Priority Case is created.

Answer:

- **Platform Event** → Subscribe via middleware (MuleSoft/Heroku) → Send Slack API call.

---

85. Scenario:

Auto-update all related Opportunities when Account's Industry field changes.

Answer:

- **After Update Flow** → Get all Opportunities → Update Industry field.
- If too many → **Batch Apex**.

---

86. Scenario:

When Contact email is updated, log change in a custom object "Email_Change_History__c".

Answer:

- **After Update Trigger** on Contact.
- Compare Trigger.old.Email vs Trigger.new.Email.
- Insert into custom object.

---

87. Scenario:

When a user uploads file to Account, auto-share with related Opportunities.

Answer:

- Use **ContentDocumentLink Trigger**.
- Create additional ContentDocumentLinks for related Opportunities.

---

88. Scenario:

Automatically deactivate users who haven't logged in for 90 days.

Answer:

- **Scheduled Apex** → Query Users by LastLoginDate.
- Update IsActive = false.

---

89. Scenario:

Update parent Case field "Child_Count__c" whenever child cases are added or removed.

Answer:

- **After Insert/After Delete Trigger** on Case.
- Aggregate child count and update parent Case.

---

90. Scenario:

Expose Account and Contacts data to external system in JSON format.

Answer:

- **Custom REST Apex Class** with @RestResource(urlMapping='/Accounts/*').
- Query Accounts + Contacts → return as JSON.