

## REMEMBER THESE

- **Personalize** = your own view. **Customize** = changes for everyone (or a role/group).
- Lists show many records; forms show one. Filters narrow lists; UI Policies change form behavior.
- **ACLs** control who can create/read/write/delete data. Roles control what users can open in the UI.
- Import path: data source → import set → transform map → target. **Coalesce** decides update vs insert.
- **CMDB** stores CIs and relationships. **CSDM** is the recommended data model for services and CIs.
- **Update sets** move config between instances. Do not develop directly in production.
- UI Policies and Client Scripts are not a security boundary — use ACLs for real access control.

## 1 PLATFORM OVERVIEW &amp; INSTANCE DOMAIN 1

| Concept             | Must know                                                                        |
|---------------------|----------------------------------------------------------------------------------|
| Instance            | Your ServiceNow environment (often separate Dev / Test / Prod).                  |
| Application         | A packaged set of features (Incident, Problem, Change, custom apps).             |
| Plugin              | Optional platform capability you activate when needed.                           |
| Now Platform        | Shared services: tables, forms, workflow, security, integrations, analytics.     |
| Single-tenant cloud | Each customer instance is dedicated; upgrades and configs are instance-specific. |

## PLATFORM CAPABILITIES

ITSM workflows, CMDB, Service Catalog, Knowledge, Flow/Workflow automation, reporting/analytics, integrations, and mobile/self-service experiences.

**Exam tip:** Know which environment you are in before making config changes. Promote tested changes with update sets or app repo paths — not ad-hoc Prod edits.

## 5 LISTS, FILTERS, TAGS &amp; FORMS DOMAIN 3

| Piece                      | Must know                                                               |
|----------------------------|-------------------------------------------------------------------------|
| List anatomy               | Title, breadcrumbs/filter, columns, rows, list controls.                |
| Filter / condition builder | AND/OR conditions; save as personal or shareable filter.                |
| Tags                       | Label records for quick find/collaboration without changing data model. |
| Form anatomy               | Header, fields, sections, related lists, activity/stream, UI actions.   |
| Form templates             | Prefill values for faster record creation.                              |
| Views                      | Different form/list layouts for different audiences.                    |

## FORM CONFIGURATION RULES

- Adding a field to a form can create a table column if the field is new.
- Do not put the same editable field in multiple form sections.
- Use sections and annotations to keep long forms usable.
- Related lists show records linked by reference/relationship.

## 9 DATA SCHEMA &amp; IMPORTING DATA DOMAIN 5

| Concept       | Meaning                                                                |
|---------------|------------------------------------------------------------------------|
| Table         | Stores records; extends another table when inheriting fields/behavior. |
| Field types   | String, Choice, Reference, Date/Time, Boolean, Integer, Journal, etc.  |
| Reference     | Points to a record on another table (many-to-one from the child side). |
| Dictionary    | Defines field properties, defaults, and attributes.                    |
| Import Set    | Staging area for inbound rows before transform.                        |
| Transform Map | Maps source fields to target; coalesce finds existing records.         |

CSV/XLSX → Data Source → Import Set → Transform Map → Target table

**Import tip:** Use Test Load 20 Records first. Bad coalesce = duplicates. Empty coalesce fields can create unexpected inserts.

## 2 NEXT EXPERIENCE NAVIGATION DOMAIN 1

| UI element | What it does                                                             |
|------------|--------------------------------------------------------------------------|
| All menu   | Search and open applications/modules.                                    |
| Favorites  | Pin frequently used modules for quick access.                            |
| History    | Recently visited lists, forms, and pages.                                |
| Workspaces | Role-focused experiences (Agent Workspace / Next Experience workspaces). |
| User menu  | Profile, preferences, impersonation (when allowed), logout.              |

- Use global search to find records, knowledge, and catalog items (based on access).
- Impersonate to verify what another user can see — then stop impersonating.
- Module visibility depends on roles; record visibility depends on ACLs/query rules.

All Favorites History Workspaces Notifications

## 6 TASKS, VTBS, ANALYTICS &amp; NOTIFICATIONS DOMAIN 3

| Area                            | Must know                                                          |
|---------------------------------|--------------------------------------------------------------------|
| Task table                      | Base for Incident, Problem, Change, and many work records.         |
| Assignment                      | Assignment group + assigned to; work notes vs additional comments. |
| Visual Task Board               | Kanban-style board for task progress and collaboration.            |
| Reports / visualizations        | Charts and lists built on table data and filters.                  |
| Dashboards / Platform Analytics | Collections of visualizations for roles/teams.                     |
| Notifications                   | When to send, who receives, what content (email/SMS/etc.).         |

**Notification tip:** Trigger + conditions + recipients + message. Test with a known user and confirm subscription/notification preferences.

- Work notes are usually internal; additional comments are often customer-visible.
- Dashboards should answer a job question, not just display charts.

## 10 ACCESS CONTROL &amp; SECURITY CENTER DOMAIN 5

| Control               | What it protects                                                                      |
|-----------------------|---------------------------------------------------------------------------------------|
| Role                  | Grants access to apps/modules and is used by ACLs/scripts.                            |
| Group                 | Assign roles and work ownership to sets of users.                                     |
| ACL                   | Table/field create, read, write, delete (and execute where applicable).               |
| Security Center       | Visibility into security posture, hardening, and admin security tools.                |
| Shared responsibility | ServiceNow secures the platform; customers secure config, access, and data practices. |

## ACL PASS LOGIC

Matching ACL requires role/condition/script to pass. Field access usually needs table + field checks. Elevate to **security\_admin** before creating ACLs. Debug with security debugging / impersonation.

create read write delete execute

## 3 APPS, PLUGINS &amp; PERSONALIZATION DOMAIN 2

| Action                    | Meaning                                                   |
|---------------------------|-----------------------------------------------------------|
| Install / activate plugin | Turns on platform features for the instance.              |
| Personalize               | User-specific changes (columns, form layout preferences). |
| Customize                 | Instance configuration that affects other users.          |
| System Properties         | Instance-wide settings that control platform behavior.    |
| Branding                  | Logo, colors, and instance identity for end users.        |

## FAST CONTRAST

If only you see the change, it is usually personalization. If everyone (or a role) sees it, it is customization/configuration.

- Activate only plugins you need; unused plugins add complexity.
- Document instance-level property changes in your change process.

## 7 KNOWLEDGE MANAGEMENT DOMAIN 4

| Concept            | Meaning                                                   |
|--------------------|-----------------------------------------------------------|
| Knowledge Base     | Container for articles with its own audience/permissions. |
| Article            | Reusable content (how-to, FAQ, known error, policy).      |
| Workflow / publish | Draft → review → published; retired when outdated.        |
| Categories         | Organize articles for browse/search.                      |
| User flags         | Helpful/not helpful and feedback improve content quality. |

## ADMIN CHECKLIST

- Control who can read vs contribute vs publish.
- Keep articles short, searchable, and tied to real incidents/requests.
- Retire stale content; do not leave conflicting published answers.

Draft Review Published Retired

## 11 CMDB &amp; CSDM DOMAIN 5

| Term             | Meaning                                                                        |
|------------------|--------------------------------------------------------------------------------|
| CI               | Configuration Item — a tracked asset/service component in the CMDB.            |
| Class / table    | CI type (Server, Application, Service, etc.).                                  |
| Relationship     | Typed link between CIs (Runs on, Depends on, Contains...).                     |
| CSDM             | Common Service Data Model — standard way to model services and supporting CIs. |
| Data foundations | Clean identification, ownership, and health of CMDB data.                      |

## ADMIN FOCUS

- Prefer CSDM-aligned classes/relationships over one-off custom models.
- Know where CIs are used: incident impact, change planning, service mapping, reporting.
- Poor CMDB quality breaks automation and service views — fix ownership and duplicates early.

## 4 COMMON PLATFORM INTERFACES DOMAIN 2

| Interface              | Use                                                             |
|------------------------|-----------------------------------------------------------------|
| List                   | Browse, filter, sort, and bulk-update records.                  |
| Form                   | Create/update one record; related lists show connected records. |
| Homepages / Dashboards | Visual summary of work and KPIs.                                |
| Service Portal / ESC   | Employee/customer self-service experience.                      |
| Workspace              | Agent-centric UI for high-volume work.                          |

## ADMIN HABIT

Configure once in a non-prod instance, test with the right roles, then promote. Always verify both navigation access and data access.

List Form Related list Dashboard Portal

## 8 CATALOG, WORKFLOW STUDIO &amp; VIRTUAL AGENT DOMAIN 4

| Component       | Must know                                                             |
|-----------------|-----------------------------------------------------------------------|
| Service Catalog | Orderable items/categories for request fulfillment.                   |
| Catalog Item    | What users request; variables collect inputs.                         |
| Record Producer | Catalog UI that creates a target record (often a task).               |
| Order Guide     | Bundles multiple items into one guided order.                         |
| Workflow Studio | Build flows/subflows/actions for automation (preferred for new work). |
| Virtual Agent   | Conversational self-service that can answer and start requests.       |

## AUTOMATION TIP

A flow has a trigger + actions/logic. A subflow has inputs/outputs and no trigger. Test in non-prod; flow tests can create real records.

- Catalog permissions and user criteria control who can see/order items.
- Fulfillment can create tasks, approvals, and notifications automatically.

## 12 AUTOMATION CONFIG, UPDATE SETS &amp; SCRIPTING DOMAIN 6

| Tool          | Must know                                                             |
|---------------|-----------------------------------------------------------------------|
| UI Policy     | Client-side mandatory/visible/read-only on forms (not security).      |
| Data Policy   | Enforces rules beyond the form (lists, imports, web services).        |
| Business Rule | Server logic on DB operations (Before/After/Async/Display).           |
| Client Script | Browser-side form logic (onLoad/onChange/onSubmit/onCellEdit).        |
| Update Set    | Captures configuration changes for migration between instances.       |
| Scripting     | GlideRecord/GlideSystem and scoped APIs; prefer Flow where practical. |

## TROUBLESHOOTING / EXAM TRAPS

| Symptom                        | Likely issue / fix                                            |
|--------------------------------|---------------------------------------------------------------|
| Module visible, records hidden | Role passed; ACL failed. Debug security rules.                |
| Works on form, not import/list | UI Policy/Client Script only. Use Data Policy or server rule. |
| Import creates duplicates      | Missing/wrong coalesce or dirty source keys.                  |
| Update set missing changes     | Wrong current update set, or change not update-set tracked.   |
| Script works in Dev only       | Missing dependency, scope/ACL, or unmigrated config.          |