DUMPSBOSS.

Building Applications with Force.com and Visualforce

Salesforce DEV-401

Version Demo

Total Demo Questions: 75

Total Premium Questions: 847

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Data Modeling	148
Topic 2, User Interface	132
Topic 3, Business Logic	156
Topic 4, Security	215
Topic 5, Reports and Dashboards	42
Topic 6, Data Management	105
Topic 7, Force.com Sites	1
Topic 8, Salesforce1 Mobile	5
Topic 9, Testing	7
Topic 10, Deployment	24
Topic 11, Mix Questions	12
Total	847

QUESTION NO: 1

You cannot relate records from the same object via an relationship field.

A. True

B. False

ANSWER: B

Explanation:

False is correct because Salesforce supports self-relationships: a relationship field can relate one record to another record of the same object. This is commonly implemented with a lookup relationship. For example, the standard Account object has the Parent Account field, which lets an account record point to another Account record. A custom object can similarly contain a lookup field targeting that same custom object, allowing hierarchical or peer-to-peer structures such as an employee's manager, a product component's parent component, or a project's parent project.

A self-relationship stores the related record's ID just like other lookup relationships and can be used in page layouts, reports, formulas, queries, and automation. When creating the field, the object is selected as both the source object and the related object. Salesforce documentation describes lookup relationships as links between records and identifies self-relationships as a supported relationship pattern. See Salesforce Developers: Object Relationships Overview and Salesforce Help: Relationship Fields.

QUESTION NO: 2

Which of the following are the 3 layers of an application? (Select all that apply)

A. User Interface

B. Business Logic

C. Business Data

D. Data Model

E. Objects

ANSWER: A B D

Explanation:

Salesforce application architecture is commonly described in three logical layers: the user interface, business logic, and data model. **User Interface** is the presentation layer through which users view information and interact with the application. In Salesforce, this can include standard Lightning pages, Lightning Web Components, Visualforce pages, and mobile experiences. **Business Logic** contains the rules and automation that apply business processes consistently, such as validation rules, flows, approval processes, formulas, and Apex. This layer determines how the application responds to user actions and data changes. **Data Model** represents the application's persistent data structure, including objects, fields, relationships, and record types. It establishes how information is stored and related across the organization. Together, these layers separate presentation, processing, and storage concerns, allowing an application to be maintained and extended without tightly coupling its interface, rules, and underlying records. Salesforce's declarative and programmatic tools support all three layers on the platform. See the Salesforce Platform Architecture Overview and the Salesforce Architecture Trailhead module.

QUESTION NO: 3

Which of these can have only one of them active at a time?

A. Validation rule

B. Assignment rule

C. Auto-response rule

D. Workflow rule

E. Escalation rule

F. Approval process

G. Sharing rule

ANSWER: B C E

Explanation:

Assignment rule, Auto-response rule, and Escalation rule are correct because Salesforce allows only one active rule of each of these rule types for the applicable object or process at a given time. An active assignment rule determines how newly created leads or cases are assigned to users or queues; although the rule can contain many ordered rule entries, Salesforce uses one active assignment rule for leads and one for cases. Similarly, an active auto-response rule can contain multiple entries that select the appropriate email template based on conditions, but only one auto-response rule is active for leads or for cases. Escalation rules apply to cases and can include several escalation actions and criteria; activating a rule makes it the single rule Salesforce uses to evaluate case escalation. This one-active-rule model provides deterministic processing while still allowing administrators to maintain alternate rules in an inactive state for later activation. Salesforce's lead assignment-rule guidance is available in Trailhead: Set Up Lead Assignment Rules. For the case-management automation model, including case assignment, auto-response, and escalation, see Trailhead: Case Management.

QUESTION NO: 4

If a user needs to give access to just one record, which feature should they use?

A. Roles

B. Role Hierarchy

C. Profile Setting

D. Manual Sharing

ANSWER: D

Explanation:

Manual Sharing is the appropriate feature when access must be granted to one specific record. It lets a record owner, a user above the owner in the role hierarchy, or an administrator share an individual record directly with a named user, role, role and subordinates, public group, or territory where applicable. The person sharing the record selects the access level, such as Read Only or Read/Write, without changing the organization-wide defaults or granting broad access to every record of that object type.

This is especially useful for an exception to the normal sharing model—for example, when an opportunity owner needs a colleague to collaborate on a single opportunity while keeping the owner's other opportunities private. The sharing entry is associated with that specific record, providing targeted access that can be reviewed or removed later. Salesforce describes manual sharing as a record-level sharing mechanism and notes that its availability depends on the object and the user's authority to share the record. See Salesforce Help: Manually Share Records and Salesforce Developer Guide: Record-Level Security and Sharing.

QUESTION NO: 5

In a Master-Detail Relationship, ownership and access to the child record are determined by the parent record

A. True

B. False

ANSWER: A

Explanation:

True is correct. In a master-detail relationship, the detail (child) record is tightly coupled to its master (parent) record for ownership and sharing purposes. The detail record does not have an independent Owner field; instead, its ownership is effectively inherited from the master record. Likewise, users' access to the detail record is controlled by their access to the master record. This inheritance model ensures that sharing and visibility remain consistent across related records, which is appropriate when a child record should not be managed, secured, or retained independently of its parent.

Master-detail relationships also affect record lifecycle behavior: deleting a master record deletes its associated detail records, and detail records can participate in roll-up summary calculations on the master. These behaviors reflect Salesforce's design that detail records are dependent on the master record. Salesforce documents that the detail record automatically inherits the sharing settings of its master record and that ownership of the detail record is determined by the master. See Salesforce: Custom Object Relationships and Salesforce Developer Documentation: Relationships Among Objects.

QUESTION NO: 6

What does development mode enable? (Select all that apply)

A. Inline editor

B. File check In and out

C. Page creation

D. Development sandbox

ANSWER: A C

Explanation:

Development mode in Salesforce is a Visualforce productivity feature intended to speed up page development directly in the Salesforce user interface. It enables an inline editor, allowing a developer to view and edit a Visualforce page's markup while working with the rendered page. This provides a quick feedback loop: after modifying the markup, the developer can save the page and immediately see the updated output without navigating separately through Setup for every change.

Development mode also supports page creation from the development interface. When enabled for a user, Salesforce exposes development tools that make it possible to create a new Visualforce page as part of the workflow for working with Visualforce. The feature is user-specific and is enabled from the user's personal settings, making it particularly useful for developers building and iterating on pages in a development environment.

Salesforce's Visualforce documentation describes development mode and its inline editing workflow in the Visualforce Developer Guide. For broader Visualforce concepts and page development guidance, see Introduction to Visualforce.

QUESTION NO: 7

Which of the following statements are true about a lookup relationship? (Select all that apply):

A. A maximum of 2 relationships are allowed per object

B. A lookup relationship can span to multiple layers

C. A parent record is required for each child

D. A lookup field is not a required field

E. Access to parent determines access to children

ANSWER: B D

Explanation:

A lookup relationship can span to multiple layers because an object that contains a lookup to another object can itself be referenced by a lookup from an additional object. This supports flexible, chained data models while keeping the related records independent. For example, a custom record can look up to an Account, and another custom record can look up to that first custom record. Lookup relationships do not impose the tightly coupled ownership and lifecycle behavior associated with master-detail relationships, so they are appropriate when related records need to remain independently manageable.

A lookup field is not a required field because, by default, Salesforce allows the lookup value on the child record to be blank. A record can therefore be created without selecting a related parent record, and an existing lookup value can generally be cleared. Administrators can apply validation rules or other business-process controls when a relationship value must be supplied, but that does not change the fundamental optional nature of a lookup relationship. Salesforce documentation describes lookup relationships as a flexible way to associate records without requiring the child record to be dependent on a parent. See Salesforce Trailhead: Object Relationships and Salesforce Help: Object Relationships Overview.

QUESTION NO: 8

If an approver rejects an approval request, then the final rejection actions are always executed.

A. True

B. False

ANSWER: A

Explanation:

True is correct. In a Salesforce approval process, final rejection actions are the actions configured to run when an approver rejects the record's approval request. A rejection ends the approval process for that request and invokes the process's configured final rejection actions. These actions can include field updates, email alerts, tasks, outbound messages, and submission of the record for another approval process, allowing an organization to consistently update the record and notify the appropriate people after a rejected request.

Final rejection actions are defined at the approval-process level rather than on an individual approval step. Consequently, they provide a standard, centralized outcome whenever the request receives a rejection during that process. Salesforce distinguishes these from initial submission actions, which run at submission, and final approval actions, which run after all required approvals are obtained. See Salesforce's Approval Process Actions documentation and the Trailhead approval processes overview.

QUESTION NO: 9

Reports can only display 1 child object at a time.

A. True

B. False

ANSWER: B

Explanation:

False is correct. Salesforce custom report types can include fields from a relationship hierarchy containing up to four objects: one primary object and up to three related objects. This permits a report to display data across multiple related child levels, rather than being limited to one child object overall. For example, a custom report type can be configured around a hierarchy such as an account, its contacts, and records related to those contacts, provided the relevant object relationships exist and are supported in the report type.

The report type determines which objects and fields are available to the report builder. When creating or editing a custom report type, an administrator defines the primary object and the related objects that form the report's relationship hierarchy. The resulting report can then use fields from each object included in that hierarchy. Salesforce documents that custom report types support a maximum of four objects in this structure, which directly establishes that reports are not limited to displaying data from only one child object. See Salesforce's Custom Report Types documentation and the Custom Report Types Trailhead module.

QUESTION NO: 10

Identify all statements that are true:

A. If you remove access to an app from a profiles, the users in that profile will still be able to see the tabs in that application.

B. If you hide a tab from a profile, the users in that profile will not be able to see records for that object.

C. If you have 2 records types for an object, you need to have 2 page layouts for that object.

D. If a user does not have access to a specific record type, they can still view records of that record type if they have access to those records.

E. Hiding a tab does not remove a user's object permissions or record-level access.

ANSWER: D E

Explanation:

Users can view a record even when its assigned record type is not available to their profile, provided that they have the required object permission and record-level access. Record type availability primarily controls which record types users can select when creating records and which associated page layouts and picklist values apply. It is not itself a record-sharing mechanism. Consequently, existing records remain accessible according to the organization's object permissions, role hierarchy, sharing rules, manual sharing, and other record-access controls. Salesforce's record type documentation describes how record types are made available to profiles and used to control business processes, page layouts, and picklist values: Salesforce Help: Record Types.

Separately, a tab's visibility is only a navigation setting; it does not grant or revoke permission to the underlying object or to individual records. Access to an object is governed by object permissions, while access to a particular record is governed by the sharing model. Therefore, hiding a tab does not prevent a user who otherwise has the necessary access from reaching records through search, related lists, reports, links, or other supported navigation paths. Salesforce documents profile-based object permissions and application/tab visibility independently: Salesforce Help: User Profiles.

QUESTION NO: 11

Apex sharing reasons are available for standard and custom objects.

A. True

B. False

ANSWER: B

Explanation:

False is correct. Apex sharing reasons, also called custom sharing reasons, are available only for custom objects. An administrator defines a custom sharing reason in the object's sharing settings, and Apex then uses the corresponding value in the object's `RowCause` field when it creates a share record, such as `MyObject__Share`. This identifies the business reason for programmatic access and makes the access easier to manage and audit.

A key benefit is that sharing records created with a custom Apex sharing reason are retained when record ownership changes. This is important when an application grants access based on a durable business rule rather than an owner's

discretionary sharing decision. Apex code should use the generated sharing-reason API name, typically in the form `Reason_Name__c`, when assigning `RowCause` for a custom-object share record.

Salesforce documents Apex managed sharing and the use of custom sharing reasons for custom objects in the Apex Developer Guide. The setup and purpose of these reasons are also covered in Salesforce Help: Define Apex Sharing Reasons.

QUESTION NO: 12

What are the 4 Declarative Application building blocks of the User Interface? (Select all that apply)

A. Visualforce Pages

B. Applications

C. Tabs

D. Workflow

E. Page Layouts

F. Record Types

ANSWER: B C E F

Explanation:

The four declarative user-interface building blocks are Applications, Tabs, Page Layouts, and Record Types. Applications provide the top-level workspace that groups related items for users, helping organize navigation around a business process. Tabs expose objects, records, Visualforce content, or other functionality in the application navigation so users can reach it without custom code. Page Layouts define the arrangement of fields, related lists, buttons, and actions shown when users view or edit records. Record Types support distinct business processes and can assign different page layouts and picklist values to different groups of users. Together, these metadata features let an administrator shape how users navigate to data and how record screens are presented through configuration rather than programmatic development. Salesforce's customization guidance describes how page layouts and record types control record experiences, while its navigation documentation explains the role of apps and tabs in organizing user access. See Salesforce Trailhead: Customize a Salesforce Object and Salesforce Help: Customize Navigation.

QUESTION NO: 13

What license is needed to install or uninstall apps from AppExchange?

A. Install package license

B. Uninstall package license

C. Manage package license.

D. Download AppExchange Packages permission

ANSWER: D

Explanation:

The required access is the **Download AppExchange Packages** permission. Although the question uses the word "license," Salesforce controls the ability to install and uninstall AppExchange packages through a system permission rather than through a separately assigned package-management license. An administrator can grant this permission through a profile or permission set to a user who must perform package installation or removal. Users with administrative access commonly receive it through their profile, but it can also be assigned more selectively when package deployment responsibilities are delegated.

This permission is relevant to managed, unmanaged, and other package installations obtained from AppExchange because package installation changes metadata and can add components, permissions, objects, and configuration to an organization. Before installing, the designated user should also review the package's access requirements and test in a sandbox when appropriate. Salesforce's package-installation guidance identifies the necessary permission and describes the installation workflow. See Install Packages and User Permissions and Access.

QUESTION NO: 14

Once a sharing rule has been created to share Account records with Public Group A with read/ write access to the shared accounts, then this sharing rule can be modified to share the Account records with Public Group B with read-only access.

A. True

B. False

ANSWER: A

Explanation:

True is correct. Salesforce sharing rules are configurable after they are created. An administrator can edit an Account sharing rule to change the group of users receiving access, such as replacing Public Group A with Public Group B. The administrator can also change the access level granted by the rule, including changing Read/Write access to Read Only access. When the edited rule is saved, Salesforce recalculates record sharing so that the rule's revised target group and access level are applied to the Account records that meet the rule criteria.

This flexibility lets an organization adapt its sharing model as team membership, responsibilities, or data-access requirements change, without needing to delete and recreate the rule. The administrator must have the appropriate setup permissions, such as Manage Sharing or equivalent administrative access, and the organization-wide defaults for Accounts must permit the use of sharing rules. Salesforce's sharing-rule documentation describes sharing rules as a way to extend record access to public groups, roles, territories, or other defined recipients and to specify the level of access granted. See Salesforce Help: Sharing Rules and Salesforce Security Implementation Guide: Sharing Rules.

QUESTION NO: 15

Which of these values can a 'if' function used in a validation rule have in either the 2nd or the 3rd parameter?

A. true

B. false

C. 1

D. 0

E. null

ANSWER: A B E

Explanation:

In a validation rule, the overall formula is evaluated as a Boolean expression: when the result is true, Salesforce displays the validation error. Therefore, an `IF()` expression used to supply that Boolean outcome can return `true` or `false` from either of its result parameters. For example, `IF(ISBLANK(AccountId), true, false)` is a valid Boolean expression for a validation rule. A null value can also be used where the formula logic permits a null result; Salesforce formula evaluation treats null as the absence of a value rather than as a numeric Boolean substitute. This is useful when a branch intentionally has no value, subject to the surrounding formula being valid for the relevant result type. Salesforce's formula-function documentation describes `IF()` as returning one of two values according to whether its logical test is true, and validation rules require formula logic that determines whether a record violates the rule. See Salesforce's formula function reference and validation rule documentation.

QUESTION NO: 16

Which of the following are true about custom objects? (Select all that apply):

A. Salesforce provides a set of custom objects that you can use to store data.

B. After you create a custom object, you need to add the user interface.

C. Custom objects come with an infrastructure including reporting, auditing and access control.

D. When you create a custom object, you get a direct access to the database.

E. Custom objects are reportable and search

ANSWER: C E

Explanation:

Custom objects come with an infrastructure including reporting, auditing and access control. When an administrator creates a custom object, it becomes a metadata-defined data model entity on the Salesforce Platform and can use platform services rather than requiring separate database infrastructure. Salesforce supports object-level permissions, field-level security, sharing settings, and audit-related capabilities such as field history tracking where configured. These capabilities allow an organization to control who can create, view, edit, or delete custom-object records and to govern access to sensitive fields and data.

Custom objects are reportable and searchable. Custom-object records can be included in reports, subject to the object's reporting configuration and relationships, enabling users to analyze the data using Salesforce reporting tools. They are also available to Salesforce search when users have access to the relevant records and fields. This native reporting and search integration is a core advantage of modeling application data with custom objects on the platform. Salesforce documents custom-object configuration and its standard platform features in Create Custom Objects and explains record-access controls in Sharing and Record Access.

QUESTION NO: 17

Among two records related by master – detail relationship, the detail record can have an owner which is different from the owner of the master record.

A. True

B. False

ANSWER: B

Explanation:

False is correct because a detail record in a master-detail relationship does not have independent ownership. Salesforce derives the detail record's ownership and sharing behavior from its master record, so the detail record cannot be assigned to a user separately from the master. This inheritance is a defining property of the relationship type and ensures that access to dependent data remains aligned with access to the parent record.

In practical terms, when the owner of the master record changes, the associated detail records follow that ownership context automatically. Detail records also inherit sharing settings from the master rather than supporting their own owner-based sharing model. This behavior helps model records that are intrinsically dependent on a parent, such as line items belonging to a parent business record. Salesforce documents that detail records in a master-detail relationship do not have an owner and that their security is controlled by the master record. See Salesforce Help: Relationship Considerations and Salesforce Object Reference: Relationships Among Objects.

QUESTION NO: 18

Identify the correct statements about dependent picklists (Select all that apply):

A. Standard picklists can be controlling fields but not dependent fields

B. The maximum number of values allowed in a controlling field is 400

C. Before defining a dependency, you should ensure that your picklist has at least 1 value

D. A custom multi-select picklist can be set at the controlling field for a dependent field

E. A field can represent both a controlling field and a dependent field.

ANSWER: A C E

Explanation:

“Standard picklists can be controlling fields but not dependent fields” is correct because Salesforce permits standard picklist fields to drive the available values in a dependent picklist, while dependent picklists must be custom picklist fields, including custom multi-select picklists. A field dependency therefore uses the controlling field’s selected value to filter the values users can select in the dependent custom picklist.

“Before defining a dependency, you should ensure that your picklist has at least 1 value” is correct because dependencies map individual controlling values to available dependent values. Picklist values must exist before an administrator can create and configure those mappings.

“A field can represent both a controlling field and a dependent field” is correct for a custom picklist. This supports a dependency chain in which one custom picklist is filtered by an earlier field and, in turn, filters the choices available in another custom picklist. Salesforce documents the eligible field types and configuration rules for dependent picklists in Define Field Dependencies. Additional implementation guidance is available in the Salesforce Trailhead module on Fields and Record Types.

QUESTION NO: 19

Sharing rules are used to further restrict access defined in the Organization-wide Default settings.

A. True

B. False

ANSWER: B

Explanation:

False is correct. Organization-wide defaults establish the baseline level of record access for users who do not own a record and have not received access through another mechanism. Sharing rules are designed to open up record access beyond that baseline by automatically granting access to defined groups of users, such as public groups, roles, roles and subordinates, territories, or portal users, based on record ownership or criteria. For example, when an object’s organization-wide default is Private, a sharing rule can grant Read Only or Read/Write access to records meeting specified conditions. Sharing rules therefore extend visibility and access; they do not reduce or further restrict the access already provided by organization-wide defaults. Salesforce’s sharing model is additive: users can gain access through ownership, role hierarchy, sharing rules, teams, manual sharing, and Apex managed sharing. Administrators should configure the organization-wide default to the most restrictive baseline appropriate for the object, then use sharing mechanisms to grant the additional access required for business processes. See Salesforce Help: Sharing Rules and Salesforce Help: Record-Level Security and Sharing.

QUESTION NO: 20

A lead can have more than one values in the lead status picklist with the ‘Converted’ checkbox checked for them so that these values are available while converting the lead to be assigned to the ‘Status’ picklist as the converted status.

A. True

B. False

ANSWER: A

Explanation:

True is correct. Salesforce lets an administrator identify multiple Lead Status picklist values as converted statuses by selecting the *Converted* checkbox for each applicable value. These statuses represent the valid final states for leads that have been converted and are available for selection in the lead-conversion process. This is useful when an organization needs to distinguish different successful conversion outcomes, such as a lead converted after qualification versus a lead converted through a partner process, while still treating each outcome as a converted lead for reporting and lifecycle purposes.

Lead Status is a standard picklist whose values are configured in Setup. The converted designation is metadata on the Lead Status value, rather than a separate custom field or a one-value-only setting. When a user converts a lead, Salesforce applies a selected converted lead status to the lead record, preserving the organization's chosen status classification after conversion. Administrators should therefore configure all required final conversion states as Lead Status values and mark each appropriate one as converted. Salesforce documents the management of lead statuses and converted-status behavior in its lead setup and conversion guidance: Customize Lead Status Values and Convert Leads.

QUESTION NO: 21

Which of the following actions are tracked in debug logs? (Select all that apply.)

A. Database changes

B. Manual workflow processes

C. Request-response HTML

D. Resources used by an Apex script

ANSWER: A D

Explanation:

Debug logs track **Database changes** because Salesforce records database activity that occurs during a traced transaction, including operations such as SOQL queries, SOSL searches, and DML actions that insert, update, delete, undelete, or merge records. These entries help a developer establish the order of execution and determine what data operations were performed by code, declarative automation, or the platform during the request.

Resources used by an Apex script are also tracked. Debug logs include execution details and governor-limit usage, such as consumed CPU time, SOQL query counts, queried rows, DML statements, DML rows, heap usage, and callouts. This information is essential when diagnosing exceptions, analyzing performance, and ensuring Apex remains within Salesforce multitenant governor limits. Log detail is controlled through debug levels, allowing developers to select the appropriate amount of Apex, database, workflow, validation, and system information for a monitored user or automated process. Salesforce documents the available debug-log categories and how to inspect execution activity in its developer documentation.

See Salesforce Debug Log Levels and Apex Governor Limits.

QUESTION NO: 22

In a workflow rule which of the following statements is true

A. Specifying at least one criterion is compulsory in the rule criteria section under Run this rule if the following criteria are met

B. Specifying criteria is optional in the rule criteria section under
Run this rule if the following criteria are met'
If no criteria are specified, then this workflow rule will execute for all records created of that object.

ANSWER: A

Explanation:

“Specifying at least one criterion is compulsory in the rule criteria section under Run this rule if the following criteria are met” is correct. A workflow rule needs defined rule criteria to determine which records qualify for the rule’s actions. When the administrator selects the criteria-based condition option, they must add one or more field criteria that Salesforce evaluates against the record. Those conditions establish the logical boundary for when the workflow rule is eligible to run, such as when a record’s status, amount, owner, or another field value meets the configured requirement.

Workflow automation evaluates the rule criteria together with the selected evaluation criteria, which determine whether the rule is considered only when a record is created, on edits, or when an edit causes a record to begin meeting the rule criteria. This separation lets an administrator precisely define both the qualifying record state and the events that can trigger the workflow action. Salesforce’s workflow-rule setup documentation describes configuring evaluation criteria and rule criteria as required parts of defining a workflow rule. See Salesforce Help: Create Workflow Rules and Salesforce Developer Documentation: Workflow Rules.

QUESTION NO: 23

Which of the following statements are true about field history tracking? (Select all that apply.)

A. it allows developer s to choose only custom fields for tracking changes.

B. It allows developers to choose up to 20 fields per object for tracking changes.

C. It does not track old and new values for long text area fields and multi-select picklists.

D. It replaces an old entry with a new entry in the History related list, any time a user modifies any of the standard

ANSWER: B C

Explanation:

Field history tracking allows an administrator or developer to select up to 20 fields on each object for auditing. The selected fields can include both standard and custom fields when those fields support history tracking, and Salesforce creates history entries when tracked values are changed. This provides a record of who made the change, when it occurred, and, for supported field types, the previous and new values. The 20-field limit is configured independently for each object, making it possible to tailor auditing to the business data that matters most on that object.

Salesforce can record that a long text area or multi-select picklist field was changed, but it does not store the old and new field values for those data types in field history. This distinction is important when designing an audit solution: history tracking can establish that a change occurred, while a separate approach may be needed if the actual before-and-after contents of these fields must be retained. Salesforce documents both the per-object field limit and the field-type limitations in its field history tracking guidance. See Salesforce Help: Track Field History for Standard Objects and Salesforce Object Reference: History Objects.

QUESTION NO: 24

Master-Detail Relationship, the parent field on the child record can be optional.

A. True

B. False

ANSWER: B

Explanation:

False is correct because a master-detail relationship requires every detail (child) record to be associated with a master (parent) record. The master field on the detail object is therefore required; a user cannot create or save the child record

without supplying a valid parent record. This requirement is fundamental to the relationship's ownership and data-integrity model: detail records do not have independent ownership, and they inherit sharing and security behavior from their master record. The required parent association also supports master-detail features such as roll-up summary fields on the master object, which calculate values from related detail records. If the business requirement is to permit child records to exist before, or independently of, a parent record, a lookup relationship is generally the appropriate relationship type because lookup fields can be configured as optional or required. Salesforce's relationship-field documentation identifies master-detail relationships as a tight parent-child relationship in which the parent controls the child's behavior and lifecycle. See Salesforce Relationship Considerations and Salesforce Developer Guide: Object Relationships.

QUESTION NO: 25

You can use the component to deploy a flow within a Visualforce page

A. True

B. False

ANSWER: A

Explanation:

True is correct. Salesforce provides the Visualforce `<flow:interview>` standard component specifically to embed and run an active flow interview inside a Visualforce page. A developer identifies the flow through the component's `name` attribute, enabling users to complete the flow directly in the context of the page rather than navigating separately to a Flow URL. The component can also exchange information with the page: Visualforce controller values can be passed into flow input variables, and output variables can be captured after the interview finishes. This makes it useful when a custom Visualforce user experience needs guided process automation, such as collecting information, creating related records, or presenting conditional screens. The flow must be activated before it can be run, and the viewing user must have the appropriate permissions to access the flow and any data it uses. Salesforce documents the `flow:interview` component as the supported Visualforce mechanism for embedding a flow interview. See the Salesforce Visualforce documentation for using flows and the flow:interview component reference.

QUESTION NO: 26

In Developer Edition, Validation rule for the 'Account Site' field (field of type text) of Account object is, ISCHANGED(Site) && OwnerId < > $User.Id to make sure that if the 'Account Site' field has been modified and the user is not the owner of the Account record, then an error message would be shown. The System Administrator has Read All, Modify All permission for the Account record. So the System Administrator will be able to modify the 'Account Site' field of the

A. True

B. False

ANSWER: B

Explanation:

False is correct. When an Account is saved, Salesforce evaluates active validation rules for that record. In this rule, `ISCHANGED(Site)` is true when the Account Site value is edited, and `OwnerId <> $User.Id` is true whenever the user performing the save is not the record owner. Therefore, if a System Administrator edits Site on an Account owned by another user, both conditions are true and the validation rule displays its configured error message.

Read All and Modify All grant broad object-level access, including the ability to view and edit records regardless of sharing and ownership. Those permissions do not cause Salesforce to skip validation-rule evaluation. A validation rule is enforced during the save operation unless its formula evaluates to false, the rule is deactivated, or the formula is explicitly designed with a bypass condition, such as one based on a custom permission. The stated formula has no administrative bypass. Salesforce documents that validation rules prevent records from being saved when their formulas evaluate to true, and that `ISCHANGED` evaluates whether a field's value changed in the current operation. See Salesforce Validation Rules and Salesforce Formula Functions.

QUESTION NO: 27

Which of the following statements about setup audit trail are true? (Select all that apply.)

A. Tracks setup changes made by developers and administrators

B. Displays 20 most recent changes

C. Tracks only the date of change

D. Tracks changes to data in custom object records

ANSWER: A B

Explanation:

Setup Audit Trail tracks changes to an organization's configuration and setup metadata, including actions performed by administrators and developers with setup access. Its entries identify information such as the date and time of the change, the user who made it, the setup area or action affected, and related details. This makes it a useful administrative control for reviewing configuration activity, investigating unexpected setup changes, and supporting governance processes.

The Setup Audit Trail page displays the 20 most recent setup changes directly in Salesforce. Administrators can also download the full setup audit trail history as a CSV file; Salesforce retains this downloadable history for up to 180 days. The combination of the on-screen recent-change list and downloadable history gives teams both immediate visibility and a broader review period for setup activity. Salesforce documents the tracked activities and available audit-history retention in its Setup Audit Trail documentation and describes how to access setup monitoring tools in Monitor Setup Changes.

QUESTION NO: 28

Which elements can be stored in folders?

A. True

B. False

C. Reports, dashboards, documents, and email templates.

ANSWER: C

Explanation:

Reports, dashboards, documents, and email templates is correct. In Salesforce, folders provide a way to organize these types of metadata and control who can access them. Report folders contain reports, while dashboard folders contain dashboards; access to those folders is an important part of determining whether users can view, run, edit, or manage the contained items. Document folders organize Salesforce Documents, which can be used in locations such as email templates and other legacy document-use cases. Email template folders similarly organize templates and support access management for users who create or use templated email.

The Salesforce metadata model represents a folder as a container for supported foldered items, including reports, dashboards, documents, and email templates. A folder is therefore not a generic storage mechanism for every Salesforce record or configuration item; it is a feature specifically associated with these supported content types. See Salesforce's Folder object reference and the Salesforce folders documentation for details about folder organization and access.

QUESTION NO: 29

While adding a contact role to an Account which of these contacts a counter available?

A. Contacts present in the related list of that Account

B. All Contacts from the Contact object.

ANSWER: B

Explanation:

All Contacts from the Contact object is correct. An account contact role is a relationship record used to identify a person's role in connection with an account, such as a decision maker, executive sponsor, or technical contact. When a user adds a contact role from an account, Salesforce lets the user select from contact records in the organization; selection is not limited to the contacts already displayed in that account's Contacts related list. This enables a contact to be related to an account through a defined role even when the contact's primary Account field points elsewhere, which is useful when one person participates in business with multiple accounts. The resulting Account Contact Role record captures the account, the selected contact, and the role, and it appears in the account's Contact Roles related list. Salesforce describes contact roles as a way to define the part a contact plays in relation to a business record and supports managing these related people separately from the standard account-contact relationship. See Salesforce's Account Contact Roles documentation and the Salesforce data-modeling overview for the distinction between records and their relationships.

QUESTION NO: 30

What tools available to move migration changes (metadata)? (Select all that apply)

A. Change Sets

B. DataLoader

C. Force.com IDE

D. Force.com Migration Tool (ANT-Based)

ANSWER: A C D

Explanation:

Change Sets, Force.com IDE, and Force.com Migration Tool (ANT-Based) are metadata migration tools in the context of the DEV-401 curriculum. Change Sets provide a point-and-click mechanism for sending selected supported metadata components between connected Salesforce organizations, such as a sandbox and its production organization. They package configuration and customization changes for upload and deployment rather than moving record data.

Force.com IDE was an Eclipse-based development environment that allowed developers to retrieve metadata from an organization, work with that metadata locally, and deploy changes back to an organization. Although this legacy product has since been retired and modern development generally uses Salesforce Extensions for Visual Studio Code, it was a valid metadata deployment tool for this exam's historical platform context.

Force.com Migration Tool (ANT-Based) uses Metadata API operations through Apache Ant scripts. It supports repeatable, source-controlled, and automated retrieval and deployment of metadata, making it useful for scripted release processes. Salesforce describes Metadata API deployment concepts at Salesforce Metadata API Developer Guide, and documents organization-to-organization deployment with Change Sets.

QUESTION NO: 31

'Status' is a picklist of the Account object having value 'Open' or 'Closed'. It is possible to create a sharing rule to share only those Accounts whose 'Status' is 'Open'.

A. True

B. False

ANSWER: A

Explanation:

True is correct. Salesforce supports criteria-based sharing rules, which automatically share records that meet specified field-value criteria with designated public groups, roles, roles and subordinates, or territories where applicable. For the Account object, an administrator can create a criteria-based sharing rule with a condition such as *Status equals Open*. Salesforce then grants the selected level of access to the target users for Account records that satisfy that criterion. This is appropriate when access should be based on a business attribute of the record rather than on ownership alone. Because Status is a picklist field with an available value of Open, it can be used as the field in the sharing-rule criteria. The rule is evaluated as Account records are created or updated, so records whose Status becomes Open can be shared automatically, while records no longer meeting the criterion cease to receive access through that particular rule. The organization-wide default sharing setting for Accounts must be sufficiently restrictive for the additional sharing access to be meaningful, since sharing rules expand record access rather than restrict it. Salesforce documents criteria-based sharing rules and their supported field criteria in its Sharing Rules documentation and Trailhead sharing-rules module.

QUESTION NO: 32

Opportunity is synced with

A. quota

B. forecast

C. quote

ANSWER: C

Explanation:

An Opportunity can be synchronized with its primary Quote. In Salesforce, users can create one or more quotes for an opportunity and designate one quote as the synced quote. When the quote is synced, Salesforce keeps supported sales information aligned between the Opportunity and that Quote, helping ensure that the proposal selected for the customer reflects the opportunity's current commercial details. This relationship is especially useful when sales teams manage products and pricing through quotes while tracking the overall deal through the opportunity record.

The synchronization behavior is based on the quote being associated with the opportunity and selected for syncing. Salesforce documentation describes creating quotes from opportunities and syncing a quote with its related opportunity so that the relevant information remains consistent during the sales process. See Salesforce Help: Quotes Overview and Salesforce Help: Sync Quotes with Opportunities.

QUESTION NO: 33

Since Apex sharing reasons are not available for standard objects, so sharing individual records of standard objects is not possible.

A. True

B. False

ANSWER: B

Explanation:

False is correct because the absence of custom Apex sharing reasons for standard objects does not prevent individual standard-object records from being shared. Salesforce provides share objects for supported standard objects, such as `AccountShare`, `OpportunityShare`, `CaseShare`, and `ContactShare`. Apex can create records in these share objects to grant a user or group a specified access level to an individual record. For example, code can insert an `AccountShare` record with the target account ID, a user or group ID, and an account access level.

Apex sharing reasons are custom `RowCause` values that developers define to identify and maintain programmatic shares. Salesforce makes these custom reasons available for custom objects, where they help preserve developer-managed access. Standard-object sharing instead uses the applicable predefined sharing causes, such as `Manual`, where supported. Therefore, sharing reasons and the ability to share a record are separate concepts: standard-object records can still receive

record-level access through their share records and Salesforce's standard sharing mechanisms. See Salesforce's Apex managed sharing documentation and the AccountShare object reference.

QUESTION NO: 34

Files tab shows files from:

A. Documents

B. Salesforce CRM Content from the libraries one is a member of and from one's personal libraries

C. Files from chatter from the groups one is a member of

ANSWER: B

Explanation:

Salesforce CRM Content from the libraries one is a member of and from one's personal libraries is correct in the Salesforce CRM Content context tested by this question. Salesforce CRM Content organizes managed content into libraries. A library provides the membership and access model that determines which users can browse, search for, and work with its files. Consequently, the Files tab provides access to content available through the libraries in which the user is a member, rather than exposing all content stored in the organization. It also includes the user's personal library, which is intended for content that the individual manages privately before optionally sharing or publishing it to an appropriate shared library. This library-based model lets organizations make approved files discoverable while retaining control over content visibility and collaboration. Salesforce's learning material on Files describes how files can be organized, shared, and accessed according to users' permissions, while the Salesforce Help documentation explains the role of libraries in Salesforce CRM Content. See Salesforce Files Basics and Salesforce Help: Libraries.

QUESTION NO: 35

If OWD is private for a custom object Position, then a user can view only those Position records which are owned by him if no one down the role hierarchy below him has created any Position record and t here is no sharing rule on Position object.

A. Always true

B. Not always true

ANSWER: B

Explanation:

Not always true is correct because private organization-wide defaults establish the baseline record access, not an absolute guarantee that a user can view only records they own. With a private default, record owners generally have access to their own Position records, and access can be extended through Salesforce's sharing mechanisms. The absence of records owned by users lower in the role hierarchy and the absence of a sharing rule do not eliminate every possible route by which access might be granted.

For example, a Position record can be shared manually with the user, shared programmatically through Apex managed sharing, or made accessible through other applicable sharing features. Additionally, users assigned object-level administrative permissions such as View All or Modify All for Position can access records regardless of ownership and the private default. Salesforce describes organization-wide defaults as the most restrictive starting point for record access and explains that additional access can be opened through sharing and elevated permissions. Therefore, the stated conditions are insufficient to conclude that the user can *only* view Position records they own.

See Salesforce Help: Organization-Wide Defaults and Salesforce Help: Object Permissions.

QUESTION NO: 36

Organization wide default (OWD) & sharing rules are applied at the

A. profile level

B. object level

C. record level

ANSWER: B C

Explanation:

object level is correct for organization-wide defaults because an administrator configures the baseline internal and external record-access setting separately for each supported object, such as Account, Contact, Opportunity, or a custom object. This baseline determines the minimum access users have to records they do not own before Salesforce evaluates additional sharing mechanisms. Salesforce describes organization-wide defaults as the most restrictive level of record access and provides these settings on an object-by-object basis.

record level is correct for sharing rules because a sharing rule grants additional access to individual records that meet the rule's criteria or are owned by members of specified roles, roles and subordinates, public groups, or territories. The rule can share those qualifying records with selected users or groups, extending access beyond the object's organization-wide default. In practice, the object-level OWD establishes the default record visibility model, while record-level sharing rules open access to relevant sets of records when business requirements call for it.

See Salesforce's documentation on Organization-Wide Defaults and Sharing Rules.

QUESTION NO: 37

Submitters can always recall approval requests.

A. True

B. False

ANSWER: B

Explanation:

False is correct because recalling an approval request is a configurable capability, not an action that is universally available to every submitter. In an approval process, the administrator can enable the setting that allows submitters to recall approval requests. When that setting is enabled, a user who submitted a record can withdraw the request while it is pending, returning the record from the approval process so that it can be edited or submitted again as appropriate. If the setting has not been enabled for the applicable approval process, the submitter does not receive the recall capability. The action also applies to an in-progress approval request; it is not a general ability to undo an approval request at any point in its lifecycle. Salesforce documents this behavior as part of configuring approval processes and their allowed submitter actions. See Salesforce Help: Configure Approval Processes and Trailhead: Configure an Approval Process.

QUESTION NO: 38

A file which is private:

A. Can always be viewed by only the owner of the file

B. Can always be viewed by the owner of the file and by any user who has the 'Modify All Data' permission.

C. Depends.

ANSWER: B

Explanation:

A private Salesforce file is still accessible to its owner, because the owner controls the file and can manage its sharing. In addition, a user who has the *Modify All Data* system permission has broad access that includes files throughout the organization, regardless of the file's ordinary sharing setting. Therefore, the statement that the file can always be viewed by its owner and by any user with the *Modify All Data* permission is correct. "Private" identifies the default access state for other users; it does not remove the owner's access or override administrative-level data access. The owner can also choose to share the file with specific people, groups, or records when business collaboration requires it, while retaining ownership. Salesforce documents that file access is controlled through file sharing and that users with elevated permissions can access files beyond normal sharing boundaries. See Salesforce's File Sharing documentation and the system permissions documentation for the role of organization-wide administrative permissions such as Modify All Data.

QUESTION NO: 39

A workflow rule is triggered based on

A) Based on criteria

B) If 'All records enter this workflow rule' checkbox is checked, then all records created of that object would cause the workflow rule to be triggered.

A. Only A is valid

B. Only B is valid

C. Both A and B are valid

ANSWER: C

Explanation:

Both A and B are valid. A workflow rule uses rule criteria to determine which records enter the rule. Those criteria can be defined with field-based conditions or with a formula that evaluates to true. When a record meets the configured entry conditions, Salesforce evaluates the workflow rule and can perform its associated actions, such as field updates, email alerts, tasks, or outbound messages.

A rule can also be configured so that all records for the relevant object enter the workflow rule rather than limiting entry through particular field criteria. In that configuration, records created for that object are eligible to trigger the rule, subject to the workflow rule's evaluation settings. This is useful when the workflow action should apply consistently to every new record of an object rather than only to a filtered subset.

Workflow rules are legacy Salesforce automation; Salesforce recommends using Flow Builder for new automation. The underlying concept remains important for existing workflow rules: record entry is controlled by the rule's configured criteria and evaluation behavior. See Salesforce's Workflow Rules documentation and the Trailhead workflow automation module.

QUESTION NO: 40 - (SIMULATION)

What is a price book?

ANSWER: See the explanation for the answer

Explanation:

A **price book** in Salesforce is a collection of products and the prices at which those products are sold.

The **Standard Price Book** contains the default standard prices for products.
Custom Price Books let an organization define different prices for the same products, such as discounted, regional, wholesale, or customer-specific pricing.
Price books are used with opportunities, quotes, and orders to select products and calculate sales amounts.

QUESTION NO: 41

The only way to test whether a time-dependent workflow rule is executing as expected is to wait.

A. True

B. False

ANSWER: B

Explanation:

False is correct because administrators can validate time-dependent workflow behavior without merely waiting for the scheduled time to pass. When a record meets the workflow rule's criteria, Salesforce places its pending time-dependent actions in the Time-Based Workflow queue. Reviewing that queue confirms that an action was scheduled, identifies the affected record, and shows the date and time at which Salesforce plans to execute it. This provides a direct way to verify that the rule evaluated as intended and that the time trigger was calculated correctly.

For practical testing, an administrator can also use test records and configure trigger conditions or record date/time values so that the scheduled action is due soon, then observe the resulting field update, email alert, task, or outbound message. If the record is changed so it no longer meets the rule criteria before execution, Salesforce removes applicable pending actions from the queue, which can also be verified during testing. The queue is therefore an important diagnostic and validation tool for time-based automation. Salesforce describes how to monitor pending actions in the Time-Based Workflow queue documentation and provides workflow automation guidance in the Salesforce Workflow documentation.

QUESTION NO: 42

What are the different ways a flow can be deployed? (Select all that apply.)

A. Custom button

B. Custom link

C. Visualforce page

D. Apex trigger

E. Field Updates

F. Custom actions

ANSWER: A B C

Explanation:

Flows can be made available to users through a custom button, a custom link, or a Visualforce page. A custom button or custom link can use a flow's URL to start a flow interview, allowing the flow to be launched from a record detail page or another configured location in the Salesforce user interface. This approach is useful when the flow should be initiated in the context of a particular record, because URL parameters can supply values to the flow where supported.

A Visualforce page is also a supported distribution mechanism because it can include the `<flow:interview>` component. That component starts and renders a flow interview directly within the page, enabling developers to incorporate flow screens into a customized Visualforce user experience and to exchange input or output values with the page as needed. These mechanisms are the classic Flow deployment methods tested in the DEV-401 context. Salesforce documents the Flow Visualforce component at Visualforce Developers Guide: Flow Components and provides guidance on making flows available to users at Distribute Flows.

QUESTION NO: 43

Which of these can be cloned?

A. Workflow rule

B. Approval process

C. Both of the above

ANSWER: C

Explanation:

Both of the above is correct because Salesforce provides cloning capabilities for both workflow rules and approval processes, allowing an administrator to reuse an existing automation configuration rather than recreate it from scratch. When a workflow rule is cloned, Salesforce creates a separate rule based on the original rule's definition. The cloned rule can then be renamed, adjusted, and activated independently, which is especially useful when similar automation is needed for related criteria or actions.

Approval processes can likewise be cloned to create a new process from an existing approval-process definition. This retains the established structure as a starting point, including the configured approval steps and actions, while allowing the administrator to tailor the cloned process to a new business scenario. Cloning helps maintain consistency across related automations and reduces setup effort. Salesforce continues to document workflow automation concepts and approval-process configuration in its official resources, including Salesforce Workflow Rules documentation and the Approval Processes Trailhead module.

QUESTION NO: 44

The workflow actions within a workflow rule execute in the order in which they are created.

A. True

B. False

ANSWER: A

Explanation:

True is correct. When a workflow rule's evaluation criteria are met, Salesforce processes the workflow actions associated with that specific rule in the sequence in which those actions were added to the rule. This predictable ordering is particularly important when a rule contains more than one field update and a later update relies on the value established by an earlier one. The action list on the workflow rule therefore represents the effective processing sequence for those actions.

This behavior applies to actions within the same workflow rule. It lets an administrator construct a controlled series of automated updates, notifications, tasks, outbound messages, or other supported workflow actions as part of one rule's execution. Workflow rules are a legacy automation feature in Salesforce, and Salesforce recommends Flow for new automation; nevertheless, understanding workflow behavior remains relevant for maintaining existing implementations and for certification questions covering the feature. Salesforce's workflow automation overview is available in Salesforce Help: Workflow Rules. For the broader transaction lifecycle in which workflow automation runs, see Salesforce Developer Guide: Order of Execution.

QUESTION NO: 45

Which of these operations on a file in Chatter leads to the generation of a private file?

A. Upload the file on the Files tab

B. Attach the file to a feed in Chatter

C. Upload (i.e. contribute) the file in Salesforce CRM Content personal library.

ANSWER: A C

Explanation:

Uploading the file on the Files tab generates a private file because the file is initially owned by the uploading user and is not made available to other users until the owner explicitly shares it. This lets a user store and manage a file in Salesforce before deciding whether to share it with particular people, groups, records, or feeds.

Uploading (contributing) the file in a Salesforce CRM Content personal library also generates a private file. A personal library is intended for files that the contributor owns and manages personally. Files contributed there remain private unless the owner subsequently shares them or otherwise makes them available through the relevant Salesforce content-sharing mechanisms. In both cases, privacy is the initial access state created by storing the file in a personal location rather than publishing it directly into a collaborative Chatter context. Salesforce's file model supports sharing files after upload through file-sharing records and related access controls. See the Salesforce documentation on file access and sharing concepts and the Salesforce sharing model.

QUESTION NO: 46

A record created meets entry criteria of multiple workflow rules. Which of the workflow rules will execute?

A. The first one

B. Randomly chosen one

C. Each one of them one after the other

ANSWER: C

Explanation:

Each one of them one after the other is correct. When a record is created or updated, Salesforce evaluates the active workflow rules associated with that object. A workflow rule whose evaluation criteria are met is eligible to execute; meeting the criteria for one rule does not prevent another matching workflow rule from being processed. Therefore, if the newly created record satisfies the entry criteria of multiple active rules, Salesforce processes the workflow actions for every applicable rule during the transaction. These actions can include immediate field updates, email alerts, tasks, outbound messages, and time-dependent actions that are scheduled according to their configured time triggers.

This behavior is important when designing automation because several rules can act on the same record. In particular, an immediate workflow field update can cause Salesforce to perform an additional update cycle and reevaluate applicable automation. Administrators should design criteria and field updates carefully so that the combined automation produces the intended outcome. Salesforce documents workflow rule evaluation within its order of execution and describes workflow rules as automation that acts when specified criteria are met. See Salesforce Order of Execution and Workflow Rules.

QUESTION NO: 47

Select the actions that can be associated with a workflow rule? (Select all that apply.)

A. Send an email

B. Send an outbound message

C. Create a record

D. Update a field

E. Create a task

F. Send an inbound message

ANSWER: A B D E

Explanation:

Workflow rules can invoke several standard workflow actions when their evaluation criteria are met. "Send an email" is supported through an email alert, which uses an email template and specified recipients to notify users or external

addresses. “Send an outbound message” is also a workflow action; it sends a SOAP message containing designated field values to an external endpoint, commonly for integration scenarios.

“Update a field” is supported through a field update action. A field update can set a value, apply a formula, or otherwise change a field on the workflow rule’s target record. “Create a task” is supported through a task action, enabling automatic assignment of follow-up work to a user, queue, record owner, or other valid recipient. These actions may be configured as immediate actions or, where supported, time-dependent actions, allowing automation to occur at a defined point after the workflow criteria are satisfied. Salesforce documents email alerts, field updates, tasks, and outbound messages as the available workflow actions. See Salesforce Help: Workflow Rules and Salesforce Developer Guide: Outbound Messaging.

QUESTION NO: 48

Which of these can edit a record during an approval process?

A. An administrator can edit the record.

B. The current approver can edit the record when approval-process settings allow it.

C. Both of the above.

ANSWER: A B C

Explanation:

An administrator can edit a record that is in an approval process, and the currently assigned approver can also edit it when the approval process’s record-editability settings permit approver edits. Salesforce approval processes can lock submitted records to prevent unauthorized changes while the request is pending, but this lock does not prevent administrators from making necessary changes. The approval-process configuration can also allow the current approver to edit the record as part of reviewing and completing the approval request. Therefore, both an administrator and the current approver are recognized edit-capable roles during approval, subject to the organization’s configured approval settings and the approver being the currently assigned individual. “Both of the above” correctly combines these two supported roles. This behavior lets organizations preserve control over records in approval while still allowing administrative intervention and, where appropriate, approver updates needed to complete the review. See Salesforce’s documentation on approval-process record editability and the Salesforce approval-process Trailhead module.

QUESTION NO: 49 - (SIMULATION)

If at the profile level, 'Password Never Expires' has been checked but at Setup > Security Controls > Password Policies if it is specified that password expires in 30 days, then which of the two would be true for users of that profile?

ANSWER: See the explanation for the answer

Explanation:

Users assigned to that profile will have passwords that never expire.

The profile-level `Password Never Expires` permission overrides the organization-wide password-expiration setting. Therefore, even though the Password Policies setting requires passwords to expire after 30 days, users whose profile has `Password Never Expires` enabled are exempt from that 30-day expiration policy.

QUESTION NO: 50

Account object has a custom field 'Tax' of type number. Some account records are present in Salesforce which do not have any value in this field. This custom field is now made a required field by editing it. The editing will fail as some records are present which do not have a value for this field.

A. True

B. False

ANSWER: B

Explanation:

False is correct. Salesforce allows an administrator to change an existing custom field's definition so that it is required even when older Account records contain no value in that field. Making the Tax number field required changes the validation behavior for records going forward; Salesforce does not need to populate the field on every existing record before saving the field-definition change. This permits an organization to strengthen data-quality rules gradually without requiring an immediate bulk update of all historical data.

After the field is configured as required, users must provide a Tax value when they create a new Account record and when they edit an existing Account record that needs to satisfy the required-field validation. Records that were already stored with a blank Tax value can remain in the database until they are updated or otherwise remediated. Administrators can use reports, list views, Data Loader, Flow, or other bulk-update tools to identify and populate legacy records if business processes require complete historical data. Salesforce documents required-field behavior and field-level configuration in its object and field documentation: Salesforce Help: Define Field Properties and Salesforce Developer Guide: Field Types.

QUESTION NO: 51

Specifying entry criteria is compulsory for:

A. Workflow rule

B. Approval process

C. Both

ANSWER: A

Explanation:

Workflow rule is correct because every workflow rule must define the conditions under which it is evaluated and allowed to execute. When creating a workflow rule, an administrator selects evaluation behavior, such as whether the rule runs only when a record is created or when it is subsequently edited, and then supplies rule criteria. Those criteria can be expressed as field-based conditions or as a formula that evaluates to true. This definition is essential because it identifies the record circumstances that trigger the workflow automation and any associated actions, such as field updates, email alerts, tasks, or outbound messages.

In the workflow-rule configuration process, the criteria are therefore a required part of making the rule meaningful and executable. The rule evaluates the record against those configured conditions whenever its selected evaluation behavior applies. Salesforce documentation describes workflow rules as automated processes that are based on specified rule criteria and evaluation criteria. Although workflow rules have largely been superseded by Flow for new automation, this is the relevant behavior for the legacy workflow-rule feature covered by this exam topic. See Salesforce's Workflow Rules documentation and the Workflow Automation Trailhead module.

QUESTION NO: 52

A single Sales Process can be assigned to multiple opportunity record types.

A. True

B. False

ANSWER: A

Explanation:

True is correct. In Salesforce, a Sales Process is a business process for the Opportunity object that determines which values are available in the Stage picklist. When configuring an Opportunity record type, an administrator selects a Sales Process and then controls the picklist values that users of that record type can see. Salesforce does not require a unique Sales Process for every Opportunity record type. The same Sales Process can be selected for multiple Opportunity record types when those record types should share the same set of sales stages, while still differing in other ways, such as page layouts, field availability, or assigned users.

This design supports reuse and consistent opportunity-stage management. For example, separate record types for direct and partner opportunities can use one shared Sales Process if both follow the same stage progression, even if each record type presents a different page layout or has different business-specific fields. A Sales Process is therefore a reusable configuration component rather than a one-to-one assignment with a record type. Salesforce documents Sales Processes as Opportunity business processes and identifies them as a record-type configuration dependency. See Salesforce Metadata API: BusinessProcess and Salesforce Trailhead: Record Types.

QUESTION NO: 53

Which of these can edit a record during an approval process?

A. Only Administrator

B. Only Current approver

C. Both of above

ANSWER: A C

Explanation:

During an approval process, Salesforce locks the submitted record to protect its integrity while approval is pending. The approval process's record editability setting determines who may edit it. An administrator can edit a locked record, which supports the "Only Administrator" configuration when organizations need strict control over changes during the approval cycle. Salesforce can also be configured to permit the currently assigned approver to edit the record, provided that person has the necessary object- and record-level edit access. Under that configuration, "Both of above" is correct because administrators retain their ability to edit and the current approver is additionally allowed to make updates. This setting is useful when an approver must correct, complete, or update information before making an approval decision. The currently assigned approver means the person who is actively responsible for the approval step; editability does not automatically extend to every prior or future approver. Salesforce documents the available approval-process record-editability choices in its approval process setup guidance: Approval Processes. More general information on the effect of record locking and approval actions is available in the Salesforce Approval Process Overview.

QUESTION NO: 54

Which of these standard fields can be tracked through Field HistoryTracking?

A. Name

B. Owner

C. Created By

D. Last Modified By

ANSWER: A B

Explanation:

Name and **Owner** can be selected for field history tracking when they are supported fields on the relevant object. Field history tracking records the prior and new values when a tracked field changes, along with the user who made the change and the date and time of that change. This creates an auditable record that users can view in the object's History related list and that developers can query through the corresponding history object, such as `AccountHistory`.

Tracking **Name** is useful because it captures changes to a record's primary business identifier. Tracking **Owner** is especially valuable for monitoring assignment and responsibility changes, such as when an account, opportunity, case, or custom record is reassigned. Salesforce lets an administrator choose supported standard and custom fields for history tracking in the object's Field History Tracking settings, subject to the organization's tracked-field limits. For more information, see Salesforce's Field History Tracking documentation and the history object reference.

QUESTION NO: 55

What type of content can be include in Visualforce pages? (Select all that apply.)

A. Text

B. HTML

C. JavaScript

D. Flash

ANSWER: A B C D

Explanation:

Visualforce pages are markup-based pages that Salesforce renders for display in a browser. They can contain ordinary **Text**, both directly in the page and as output produced by Visualforce components or expressions. They also support standard **HTML**, enabling developers to structure content with browser-recognized elements alongside Visualforce tags and components. Client-side **JavaScript** can be included to implement browser interactions, validation, dynamic behavior, and integrations with supported Salesforce JavaScript APIs; when using scripts, developers should follow Lightning Web Security and Content Security Policy requirements applicable to the page's runtime context. Visualforce can also include browser-rendered embedded media content such as **Flash**, historically through standard HTML embedding approaches. Although modern browsers have ended support for the Flash Player and new development should use modern web technologies instead, Flash is a type of content that a Visualforce page could include under the broad Visualforce content model tested by this question. Salesforce describes Visualforce as a framework for creating custom user interfaces using markup, standard web technologies, and server-side controllers. See the Visualforce Developer Guide introduction and Salesforce's Content Security Policy guidance.

QUESTION NO: 56

Which of the following statements are true? (Select all that apply.)

A. The ISCHANGED function compares the value of a field with its previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.

B. The ISNUMBER function determines if a text value is a number and returns TRUE if it is; otherwise, it returns FALSE.

C. The ISNEW function compares a text field to a regular expression

ANSWER: A B

Explanation:

The ISCHANGED function is correctly described because it evaluates whether a field's current value differs from its prior value. It returns TRUE when the value has changed and FALSE when the current and previous values are equivalent. This makes it useful in contexts that support change-detection functions, such as workflow rules, validation rules, assignment rules, escalation rules, and field updates.

The ISNUMBER function is also correctly described. It tests whether a supplied text value can be interpreted as a number. When the text represents a valid numeric value, such as `"42"` or `"3.14"`, the function returns TRUE; otherwise, it returns FALSE. This is useful when a formula must validate or branch based on whether text-based input contains numeric content before applying numeric conversion or processing logic.

Salesforce documents these functions in its formula-function reference, including their supported contexts and return behavior. See the Salesforce Formula Operators and Functions reference and the Salesforce Developer function reference.

QUESTION NO: 57

Files can be uploaded on Chatter from:

A. Files tab

B. Documents tab

C. Local file system of your computer

D. Attachments from the 'Notes and Attachments' related list

ANSWER: A C

Explanation:

Chatter supports sharing files that are already managed as Salesforce Files as well as uploading a new file directly from the user's computer. The **Files tab** provides access to files stored in Salesforce, allowing a user to select and share an existing file in the appropriate Chatter context. This supports reuse of centrally stored content while retaining Salesforce Files features such as sharing controls, versioning, previews, and collaboration activity.

A user can also select a file from the **local file system of their computer** when creating a Chatter post or adding a file through the Chatter interface. Salesforce uploads that selected local file and stores it as a Salesforce File, after which it can be shared with users, groups, or records according to the applicable access settings. These are the two supported sources identified by the Chatter file-sharing workflow. Salesforce's file documentation describes uploading files from a computer and working with files in Salesforce: Salesforce Help: Files Overview. For platform-level details on uploading and associating Salesforce Files, see Salesforce Developer Documentation: Work with Files.

QUESTION NO: 58

Identify the features of a workflow rule. (Select all that apply.)

A. It triggers an action when a record meets the criteria for the rule.

B. It can trigger only immediate actions.

C. It is evaluated before the rule is created.

D. It can be triggered on import of data.

ANSWER: A D

Explanation:

A workflow rule evaluates records against defined criteria and, when those criteria are satisfied, initiates the workflow actions configured for the rule. This can include actions such as field updates, email alerts, tasks, and outbound messages. The action can occur immediately when the record meets the evaluation criteria or, where applicable, be scheduled for a later time relative to a date field; the key feature is that the rule automates a defined business process after its criteria are met.

Workflow rules are also evaluated when records are created or updated through supported data-loading mechanisms. Consequently, importing records can cause applicable workflow rules to run, just as record creation or updates through the Salesforce user interface can. This lets administrators apply consistent automation to bulk-loaded data, provided that the imported records meet the rule criteria and the relevant workflow configuration is active. Salesforce documentation describes workflow rules as automation that evaluates records and executes associated actions, and notes that workflow can be triggered by record operations, including API-based changes used by import tools. See Salesforce Workflow documentation and Salesforce Data Loader documentation.

QUESTION NO: 59

Which of the following are the 3 layers of an application? (Select all that apply)

A. User Interface

B. Business Logic

C. Business Data

D. Data Model

ANSWER: A B D

Explanation:

Salesforce applications are commonly understood through three architectural layers: the user interface, business logic, and data model. **User Interface** is the presentation layer that users interact with, including Lightning record pages, apps, custom Lightning components, Visualforce pages, and standard Salesforce screens. It presents information and provides actions while remaining separate from the rules that govern the application.

Business Logic is the layer that implements an organization's processes and rules. In Salesforce, this can be declarative automation such as validation rules, flows, approval processes, and formula fields, as well as programmatic automation such as Apex. This layer determines how the application behaves when users or integrations create, update, or otherwise process records.

Data Model is the persistence layer. It defines how business information is structured and related through standard objects, custom objects, fields, relationships, record types, and related metadata. Objects are therefore components within the data model rather than a separate application layer. Designing these layers separately helps make applications easier to maintain, extend, and secure. Salesforce's application-development material describes building the data model, business logic, and user interface as the core elements of an app. See Salesforce application architecture guidance and Salesforce data modeling guidance.

QUESTION NO: 60

While inserting/updating/upserting a new user using the Apex Data Loader, values for which of these fields can be provided in the input csv file?

A. User Role

B. User License

C. User Profile

ANSWER: A C

Explanation:

User Role and **User Profile** can be supplied when users are loaded through Data Loader. The CSV maps these logical fields to the User object's relationship ID fields: `UserRoleId` for the role and `ProfileId` for the profile. Values must be Salesforce record IDs, rather than the displayed role or profile names. A profile is a key part of creating a user because it establishes the user's baseline permissions and settings. A role may also be assigned to place the user in the organization's role hierarchy, which supports record-access behavior such as hierarchy-based sharing. Data Loader supports mapping writable User fields during insert, update, and upsert operations, subject to the executing administrator's permissions and the organization's user-management constraints. Before loading, obtain the relevant Profile and UserRole record IDs with a report, SOQL query, or an export, then map them to the appropriate CSV columns. Salesforce's User object reference identifies `ProfileId` and `UserRoleId` as relationship fields on User, and Salesforce's Data Loader documentation describes mapping CSV columns to object fields for supported data operations. See the Salesforce User object reference and the Data Loader documentation.

QUESTION NO: 61

Time-Dependent workflow can be used when a workflow rule is set to evaluate Every time a record is created or updated.

A. True

B. False

ANSWER: B

Explanation:

False is correct because time-dependent workflow actions are not available for a workflow rule evaluated every time a record is created or updated. In the legacy Workflow Rules feature, scheduled actions require an evaluation setting that supports a defined transition into the rule criteria: either when the record is created, or when it is created and any time afterward that it subsequently meets the criteria. This lets Salesforce identify the point from which to calculate the scheduled action's time trigger and manage the pending action appropriately.

When a rule evaluates on every edit that meets the criteria, an update can repeatedly satisfy the rule without a qualifying transition. That evaluation model is therefore limited to immediate workflow actions and cannot be used to schedule time-dependent actions. Although Salesforce now generally recommends Flow Builder for new automation, this is the applicable behavior for the Workflow Rules functionality tested by the question. Salesforce documents the evaluation criteria and available actions in its workflow-rule guidance; see Time-Dependent Workflow Actions and Workflow Rules.

QUESTION NO: 62

Identify the correct statements about dependent picklists (Select all that apply):

A. Standard picklists can be controlling fields but not dependent fields

B. The maximum number of values allowed in a controlling field is 400

C. Before defining a dependency, you should ensure that your picklist has at least 1 value

D. A custom multi-select picklist can be set at the controlling field for a dependent field

E. A field can be both a controlling field and a dependent field.

ANSWER: A C E

Explanation:

Dependent picklists use a controlling field to filter the values available in another picklist. Standard picklists can be controlling fields but not dependent fields is correct because Salesforce allows standard picklist fields to drive dependencies, while dependent fields must be custom picklists. Before defining a dependency, you should ensure that your picklist has at least 1 value is also correct: the controlling and dependent picklists need defined values so that value mappings can be configured in the dependency matrix. A field can be both a controlling field and a dependent field is correct when used in separate dependencies. For example, one custom picklist can depend on an upstream controlling picklist while also controlling the choices presented by a downstream dependent picklist. This supports multi-level filtering of values. Salesforce documents the supported field types and configuration requirements for dependent picklists in its field dependency documentation. Additional implementation guidance is available in the Trailhead picklist administration module.

QUESTION NO: 63

Files tab shows files from:

A. Documents

B. Salesforce CRM content from the libraries one is a member of from one's personal libraries

C. Files from chatter from the groups one is a member of

ANSWER: B C

Explanation:

The Files tab is a consolidated workspace for content that a user can access in Salesforce. It includes Salesforce CRM Content items available through libraries to which the user belongs, as well as items in the user's personal library. Library membership and permissions determine whether CRM Content is visible, so the tab can surface centrally managed material such as presentations, documents, and other published content without requiring the user to navigate separately to each library.

The Files tab also brings together files shared through Chatter, including files posted in Chatter groups of which the user is a member. This gives users a single place to find collaborative files associated with group conversations and activities, subject to the group's membership and sharing access. Salesforce Files is designed to make files easier to discover, share, and collaborate on across Salesforce, while preserving the access controls that apply to the relevant library or Chatter group. See Salesforce's Files upload and sharing guidance and its Chatter Groups module for the related file-sharing and group-access concepts.

QUESTION NO: 64

How many versions of a flow can be active at one time?

A. 1

B. 2

C. 3

D. 4

ANSWER: A

Explanation:

1 is correct. A Salesforce flow can have multiple saved versions, which lets administrators and developers revise, test, and preserve prior iterations of the automation. However, for a given flow, Salesforce permits only one version to be activated at a time. Activating a newer version makes that version the one used when the flow runs; the previously active version is no longer active, while remaining available as an older saved version that can be viewed, cloned, or reactivated if needed.

This one-active-version model ensures that each flow has a single, unambiguous runtime definition. It also supports controlled change management: build and test a new version, then activate it when ready without losing the earlier implementation. Salesforce documents that only one version of a flow can be active at a time and that activating a version deactivates the formerly active version. See Salesforce Help: Build Flows and Salesforce Metadata API: Flow.

QUESTION NO: 65

If a lookup field is optional, what actions can you define for the field if someone deletes a referenced lookup record? (Select all that apply)

A. Clear the value of this field

B. Delete all records which contain this reference

C. Don't allow deletion of the lookup record that's part of a lookup relationship

D. Delete this record also

ANSWER: A C D

Explanation:

For an optional lookup relationship, Salesforce lets the administrator define the deletion behavior that should apply when the record referenced by the lookup is deleted. **Clear the value of this field** preserves the record containing the lookup, while removing its reference to the deleted record. This is useful when the related record is not essential to the child record's continued existence. **Don't allow deletion of the lookup record that's part of a lookup relationship** enforces referential protection: Salesforce prevents deletion while records still point to that lookup record, requiring users to first remove or change those references. **Delete this record also** creates cascading deletion behavior for the records that contain the lookup value, so dependent records are removed along with the referenced record. These choices allow a lookup relationship to support the business-appropriate balance between retaining independent records, protecting related data from deletion, and automatically removing dependent data. Salesforce documents that lookup relationships can be configured to clear references, restrict deletion, or delete related records when the referenced record is deleted. See Salesforce Object Reference: Relationships Among Objects and Salesforce Trailhead: Object Relationships.

QUESTION NO: 66

Building your Data Model True or False The limit on the number of custom fields per object depends on the Salesforce edition.

A. True

B. False

ANSWER: A

Explanation:

True is correct. Salesforce enforces a maximum number of custom fields that can be created on an object, and that maximum varies according to the organization's Salesforce edition. For example, lower editions provide a smaller custom-field allowance than Enterprise, Unlimited, and Developer editions. This is an edition-level platform limit, so it affects how a data model should be planned: before adding many attributes directly to an object, architects should verify the applicable limits and consider appropriate modeling approaches for data that may grow substantially.

The limit applies to custom fields on both standard and custom objects, subject to Salesforce's documented object and field limits. Some field types and platform features can also have their own related limits, but the central point remains that the total custom-field capacity available for an object is determined in part by the Salesforce edition assigned to the org. Salesforce publishes its current limits in its developer documentation; because limits can change over time, consult the current reference when designing or reviewing a solution.

See Salesforce's Platform Limits reference and custom field limits documentation.

QUESTION NO: 67

Interviewers should never be able to view a candidate's social security number. Which tool would you use to meet this requirement?

A. Record types

B. Field-level security

C. Page Layouts

D. Page Layouts with Record Types

ANSWER: B

Explanation:

Field-level security is the appropriate tool because it controls whether users, based on their profile or permission sets, can access an individual field on an object. To ensure interviewers can never view a candidate's Social Security number, the

administrator can set that field to be hidden for the interviewer profile and avoid granting field read access through any permission set. This protection applies consistently across the Salesforce user interface, record pages, related lists, reports, list views, search results, and API access. In other words, the field is not merely removed from one screen; it is unavailable to those users throughout the platform.

This is especially important for sensitive personally identifiable information, such as Social Security numbers, because page-level configuration alone does not provide complete data-access control. Salesforce evaluates object permissions and field-level security when determining whether a user may read or edit a field. Administrators can configure field accessibility from Object Manager or by using profile and permission-set field permissions. See Salesforce's field-level security documentation and the guide to field permissions in profiles and permission sets.

QUESTION NO: 68

In Salesforce Mobile, a user profile like Custom – Recruiter can be assigned only one Mobile Configuration at a time.

A. True

B. False

ANSWER: A

Explanation:

True is correct. In the legacy Salesforce Mobile administration model, a mobile configuration defines the mobile experience made available to users, including the objects, tabs, and navigation items that appear in the Salesforce mobile client. Administrators assign a mobile configuration to one or more profiles so that users with those profiles receive the intended mobile setup. However, each profile can be associated with only one mobile configuration at a time. This creates a clear, deterministic configuration for every user of that profile: when the user accesses Salesforce Mobile, Salesforce can apply the single configuration assigned to the user's profile rather than attempting to combine settings from multiple configurations. If different groups require different mobile experiences, the administrator should use separate profiles, or revise the profile and its assigned configuration as appropriate. This behavior is part of the older Salesforce Mobile configuration feature referenced by DEV-401-era material; modern Salesforce mobile navigation is generally managed through Lightning apps and mobile navigation settings. See Salesforce's mobile documentation at Salesforce Help: Set Up the Salesforce Mobile App and the platform release documentation archive at Salesforce Mobile SDK Developer Guide.

QUESTION NO: 69

Records can enter a step in an approval process based on

A. Based on criteria

B. If 'All records enter this step' checkbox is checked, then all records which have entered this approval process, would enter this step of the approval process

C. Only A is valid

D. Only B is valid

E. Both A and B are valid.

ANSWER: E

Explanation:

The combined statement that records may enter an approval step based on criteria or because the all-records setting applies is correct. In Salesforce, each approval step has step criteria that determine whether a record entering the approval process should enter that particular step. An administrator can define field-based conditions for the step, so only records meeting those conditions proceed to its assigned approver, approval request, and any step actions.

Alternatively, a step can be configured so that all records entering the approval process enter that step. This is done by selecting the setting commonly presented in the approval-step setup interface as “All records should enter this step.” With that setting enabled, Salesforce does not require the record to satisfy separately defined step criteria; every record that has entered the approval process is routed through the step. This configuration is useful for a universal approval stage, such as a required manager review that should occur for every submitted record. Salesforce evaluates approval-process entry criteria first to decide whether a record can enter the process, then evaluates each step’s criteria or all-records configuration to determine the route through the process. See Salesforce’s Approval Processes documentation and approval step setup guidance.

QUESTION NO: 70

For manual sharing of any account record, by default the owner of the record has Full Access to the account record and has Read/Write access on the related contacts, opportunities, cases. The Read/Write access on the related contacts, opportunities, cases can be made more restrictive for the owner like read-only by creating a new manual sharing rule for that account record and selecting the owner's name from within the Users list.

A. True

B. False

ANSWER: B

Explanation:

False is correct because manual sharing is an additive mechanism: it grants access to a record and, where applicable, to related records; it does not reduce, override, or revoke access that a user already has. An account owner’s access is derived from ownership, and the owner retains full access to the Account record. Access to related Contacts, Opportunities, and Cases can also be supplied through account-related implicit sharing and the organization’s sharing configuration. Creating a manual share for the owner cannot downgrade that existing access from Read/Write to Read Only.

Salesforce record sharing determines the greatest level of access available to a user when more than one source of access applies. Consequently, adding a share entry for the owner with a lower access level cannot make the owner’s effective access more restrictive. To limit access, an administrator must instead use appropriate ownership, organization-wide defaults, sharing settings, role hierarchy behavior, and object permissions; manual sharing itself is not a deny or restriction tool. Salesforce documents that record owners have full access to their records and that sharing rules extend record access. See Salesforce Record Sharing Basics and Manually Share Records.

QUESTION NO: 71

Which users can grant sharing privileges on a given record? (Select all that apply)

A. System Administrators

B. Manager

C. Owner of the record

D. Users above the owner of the record in the role hierarchy

E. Users below the owner in the role hierarchy

ANSWER: A C D

Explanation:

Salesforce record sharing is controlled by record-level access and the role hierarchy. **System Administrators** can grant sharing privileges because they have broad administrative access, including the ability to manage access to records across the organization. The **Owner of the record** can manually share that record with users, roles, groups, or territories when manual sharing is available for the object. This enables the owner to give another user the appropriate level of record access without changing organization-wide defaults or ownership.

Users above the owner of the record in the role hierarchy can also grant sharing privileges. The role hierarchy is designed to provide users higher in the hierarchy access to records owned by users below them, and that higher-level access permits them to manage sharing for those records. This supports managerial visibility and controlled collaboration while retaining the existing record owner. Salesforce describes how record ownership, hierarchy access, and sharing mechanisms work in its Security Implementation Guide and its record-level security Trailhead module.

QUESTION NO: 72

What are ways in which you can obtain a record's ID? (Select all that apply.)

A. URL

B. Reports

C. Data Loader

D. On each record

E. Import Wizard

ANSWER: A B C

Explanation:

A record's Salesforce ID can be obtained from the URL, from reports, and through Data Loader. When a user opens a record, the record ID is included in the browser address; Salesforce IDs are typically 15 characters in the user interface and can be represented in an 18-character, case-insensitive form for API and integration use. This makes the URL a practical way to identify an individual record while viewing it.

Reports can expose record identifiers by including the applicable Record ID field in the report output, allowing IDs to be viewed, filtered, or exported alongside other record data. This is particularly useful when IDs are needed for a selected group of records rather than for only one record. Data Loader can also retrieve IDs when exporting records. Its export operations return the Id field with queried records, providing a reliable method for collecting IDs in bulk for updates, deletes, integrations, or data maintenance.

These approaches align with Salesforce's supported UI, reporting, and data-management capabilities. See Salesforce's Data Loader documentation and report field documentation.

QUESTION NO: 73

Which of these can be cloned?

A. Workflow rule

B. Approval process

C. Both of above

ANSWER: C

Explanation:

Both of the above is correct because Salesforce supports cloning for both workflow rules and approval processes. Cloning is a configuration-reuse feature: an administrator can start with an existing automation definition and create a separate copy to adapt for a related business requirement, rather than recreating its setup manually. For a workflow rule, the copied definition provides a starting point for its evaluation criteria and associated workflow actions. For an approval process, cloning creates a new approval-process definition based on the existing process, allowing its submission settings, approval steps, assigned approvers, and actions to be reviewed and adjusted for the new use case. The clone is a distinct metadata configuration, so it can be modified independently of the source after it is created. This capability is especially useful where processes differ only by object criteria, routing, or notifications. Although Salesforce now directs new automation development toward Flow,

workflow rules and approval processes are the relevant declarative tools for this DEV-401 question, and both include cloning support. See Salesforce's Workflow documentation and Approval Processes documentation.

QUESTION NO: 74

Identify all statements that are true:

A. If you remove access to an app from a profiles, the users in that profile will still be able to see the tabs in that application.

B. If you hide a tab from a profile, the users in that profile will not be able to see records for that object.

C. If you have 2 records types for an object, you need to have 2 page layouts for that object.

D. If a user does not have access to a specific record type, they will still be able to see records of that record type if they otherwise have record access.

ANSWER: A D

Explanation:

Removing an app's visibility from a profile controls whether users can select or navigate to that app; it does not itself change the profile's tab settings or the user's object permissions. Tabs are configured separately in profile settings, and a tab that remains visible can still be available through the user interface, such as the App Launcher or an applicable navigation configuration. Salesforce therefore treats app access, tab visibility, and object access as distinct controls. See Salesforce's guidance on customizing apps and profile tab settings.

Record-type availability also has a different purpose from record-level visibility. Assigning record types to a profile determines the record types a user can choose when creating records and, where applicable, when changing a record type. It does not act as a sharing or visibility rule for existing records. Access to existing records remains governed by object permissions, ownership, sharing rules, role hierarchy behavior, and other record-access mechanisms. Consequently, a user can still see an existing record associated with a record type that is not available to that user, provided the user otherwise has access to that record.

QUESTION NO: 75

Which statement is true?

A. Child records in mater-detail relationships have their own org-wide defaults.

B. Org-wide defaults can be set for both standard and custom objects.

C. Only read/write access can be granted through sharing rules.

D. Sharing rules are used to restrict access to records.

ANSWER: B

Explanation:

Org-wide defaults can be set for both standard and custom objects. Org-wide defaults establish the baseline level of record access for users who do not own a record and do not receive additional access through role hierarchy behavior, sharing rules, teams, manual sharing, or other sharing mechanisms. Salesforce supports configuring these defaults for many standard objects, such as accounts, contacts, opportunities, cases, and leads, as well as for custom objects. The available baseline settings depend on the object and can include private, public read only, public read/write, or controlled by parent where applicable. Administrators configure these settings from Sharing Settings, allowing the organization to define a secure default access model before selectively opening access where business processes require collaboration. This approach is central to Salesforce's record-level security model: begin with the most restrictive access appropriate for the organization, then use sharing features to grant additional access to appropriate users or groups. Salesforce documents that org-wide defaults specify the default internal access users have to records they do not own for each object. See Salesforce Help: Organization-Wide Defaults and Salesforce Security Implementation Guide: Data Access.