DUMPSBOSS.

Salesforce Certified Platform App Builder

Salesforce Platform-App-Builder

Version Demo

Total Demo Questions: 72

Total Premium Questions: 721

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Salesforce Fundamentals	11
Topic 2, Data Modeling and Management	235
Topic 3, Security	73
Topic 4, Business Logic and Process Automation	147
Topic 5, User Interface	132
Topic 6, App Development	26
Topic 7, Mobile	24
Topic 8, App Deployment	73
Total	721

QUESTION NO: 1

An app builder wants to create a formula field on an Account to include data from related Contacts but is unable to find the relationship in the formula editor. What is a limitation of formulas that could be causing the issue?

A. Control and Account objects: 0.00 or 1 have a Master-Detail Relationship.

B. Store Data: 3000 characters in the formula.

C. Formula field that reached on the Account object.

D. unable to reference the child records.

ANSWER: D

Explanation:

Unable to reference the child records is correct because Salesforce cross-object formulas traverse relationships from a record to its parent records, not from a parent record to its collection of child records. An Account can have many related Contact records, so there is no single Contact value that an Account formula field can select or display through the formula editor. This behavior is intentional: formula fields calculate a value for the current record and can access fields through parent relationship paths, such as accessing an Account field from a Contact formula. They cannot iterate through, aggregate, or otherwise directly query the Contacts related to an Account. To place derived Contact information on an Account, the app builder should use an appropriate declarative automation or aggregation design, depending on the requirement—for example, a record-triggered flow to maintain a value on Account, or a roll-up approach where the relationship and aggregation requirements support it. Salesforce's cross-object formula guidance describes formulas as referencing parent-object fields, while its formula-field learning material explains how formulas calculate values on individual records. See Salesforce Help: Cross-Object Formulas and Trailhead: Formula Fields.

QUESTION NO: 2

Each workflow rule applies to a single object.

A. True

B. False

ANSWER: A

Explanation:

True is correct. A workflow rule is configured in the context of one specific Salesforce object, such as Account, Case, Contact, Opportunity, or a custom object. Its evaluation criteria are assessed when records of that selected object are created or updated, and its associated workflow actions—such as email alerts, field updates, tasks, and outbound messages—are executed from that object-specific rule. This object-level design gives the rule a defined record context for evaluating field values, determining whether criteria are met, and selecting related recipients or values for actions. A workflow rule can use information from related records in its criteria where Salesforce supports those references, but the rule itself remains attached to and evaluates records for one primary object only. Salesforce documents workflow rules as process automation configured for records on an object, with criteria and actions associated to that object. For additional details, see Salesforce's Workflow documentation and the Workflow Rules Trailhead module.

QUESTION NO: 3

You have created a workflow rule to send an e-mail in your configuration sandbox.

For some reason it's not working, what should you double check? (Choose two.)

A. Look at the system audit trail.

B. HTML does not work in sandbox, make sure your e-mail has no HTML.

C. You have the correct e-mail address.

D. Check the deliverability settings.

ANSWER: C D

Explanation:

You should confirm that the workflow email alert uses the correct email address and review the sandbox's email deliverability settings. An email alert can be configured with recipients sourced from a record's email field, a related user, role, public group, or an explicitly specified address. If the intended recipient address is missing, outdated, blank on the triggering record, or not the address configured in the alert, the message will not reach the expected mailbox. Confirming the address and the email-alert recipient configuration is therefore an essential troubleshooting step.

Sandbox email behavior is also controlled by Deliverability. Salesforce commonly restricts sandbox outbound email to system email only, particularly after a sandbox is created or refreshed. For workflow email alerts to be delivered to external recipients during testing, the environment must permit email through an appropriate access level, such as "All Email." Administrators can review this setting in Setup under Deliverability. Salesforce also notes that sandbox email may be redirected or otherwise managed for safe testing, so checking this org-level setting is necessary before concluding that the automation failed. See Salesforce's Email Deliverability documentation and Email Alert documentation.

QUESTION NO: 4

Northern Trail Outfitters (NTO) has created the custom objects Trail and Park in Salesforce to track trails and parks respectively. NTO wants to Track the total number of trails a park has on the park record without writing any code. Which two actions should an app builder take to accomplish this requirement? Choose 2 answers

A. Use a formula field on the Park record to the total number of trails.

B. -Use a roll-up summary field on the Park record to the total number of Trails.

C. Use a master-detail relationship between the Park and Trail objects.

D. Use a lookup relationship between the Park and Trail objects.

ANSWER: B C

Explanation:

Use a master-detail relationship between the Park and Trail objects, with Park as the parent and Trail as the child, so Salesforce maintains the parent-child relationship required for native roll-up summaries. Then, create a roll-up summary field on Park that uses the *COUNT* summary type to count its related Trail records. The roll-up value is stored on each Park record and is automatically recalculated as Trail records are created, deleted, restored, or moved to a different Park. This provides a declarative, no-code total directly on the Park record, satisfying the requirement without Apex, Flow, or an external package. Salesforce supports roll-up summary fields only when the summarized records are related through a master-detail relationship. The Trail object should therefore contain the master-detail relationship field pointing to Park, and the Park object can expose the resulting count field on its page layout or Lightning record page. For implementation details, see Salesforce's Roll-Up Summary Fields documentation and its relationship field considerations.

QUESTION NO: 5

What are use cases for Validation Rules?

A. Enforce conditionally required fields

B. Enforce proper data format

C. Enforce consistency

D. Prevent data loss

E. All of the above

ANSWER: E

Explanation:

All of the above is correct because validation rules evaluate a formula whenever a user attempts to save a record and display an error when the formula evaluates to true. This makes them a flexible declarative control for protecting data quality at the point of entry or update. A validation rule can make a field conditionally required, such as requiring a close reason only when an opportunity is closed lost. It can enforce an expected format or permitted value pattern, such as requiring an identifier to have a specified length or ensuring related field values meet defined business requirements. It can also maintain consistency across fields and records by requiring values to align with an organization's business rules. Finally, validation rules can prevent loss of important existing data during edits by blocking a save when a user clears, changes, or updates protected information in a way that violates the rule. Because these requirements can be expressed through formulas, error messages, and error locations, validation rules provide a centralized way to stop invalid changes before they are committed to the database. Salesforce describes validation rules and their save-time behavior in the Validation Rules documentation and provides practical configuration guidance in the Trailhead validation rules module.

QUESTION NO: 6

A business user at Universal Containers wants to update an Account directly from an Opportunity record.

What should the app builder create to allow the business user to make these edits?

A. An update record action with a related record component.

B. An update record action with a details component

C. Formula fields displaying the Account fields.

D. Opportunity fields updated by a process.

ANSWER: A

Explanation:

An update record action with a related record component is correct because it provides a guided, record-contextual way for users to edit fields on the Account that is related to the current Opportunity. The action is placed in the Opportunity user interface, while the related-record configuration supplies the relationship context needed to target the associated Account rather than the Opportunity itself. When a user opens an Opportunity, they can invoke the action and edit the exposed Account fields without navigating away to the Account record first. This is appropriate when the required Account edits are known and can be presented in an action layout, helping keep the user experience focused and reducing extra navigation. The action should be configured to update the related Account record and include only the Account fields that business users need to maintain; normal field-level security and object permissions still control whether a user can view or edit those fields. Salesforce quick actions are designed to make common record operations available directly in the context where users work. See Salesforce's Quick Actions overview and the Customize Quick Actions Trailhead module.

QUESTION NO: 7

An app builder has been requested to extend record access beyond the organization-wide defaults configured. Which two features satisfy this requirement? (Choose two.)

A. Using Sharing Rules.

B. Using Manual Sharing Rules.

C. Using Dynamic Role Hierarchy.

D. Using Public Groups.

ANSWER: A B

Explanation:

Using Sharing Rules and Using Manual Sharing Rules extend users' access to records beyond the baseline established by organization-wide defaults. Sharing rules provide an automated, declarative way to grant additional record access to specified roles, roles and subordinates, public groups, territories, or groups of users when record ownership or criteria-based conditions are met. They are used to open access horizontally to records that would otherwise remain private or read-only under the organization-wide default setting.

Manual sharing enables a record owner, a user above the owner in the role hierarchy, or an administrator to grant access to a specific individual record. This is appropriate when access is needed as an exception rather than through a broad, reusable rule. The sharing action creates a record-level share that supplements the organization-wide default without changing the default access model itself. Salesforce's sharing architecture combines organization-wide defaults with mechanisms such as sharing rules and manual sharing to provide the required additional access. See Salesforce Sharing Architecture and Manually Share Records.

QUESTION NO: 8

Universal Containers wants to streamline its data capture process by linking fields together. They wish to do this so that the available value on dependents fields are driven by value selected on controlling fields. Which consideration supports the stated requirements? Choose 3 answers

A. The import wizard only allows value to be imported into a dependent picklist if they match the appropriate controlling field

B. Custom picklist field can be either controlling or dependent field

C. Multi select picklist can be dependent picklist but not controlling fields

D. Standard and custom picklist fields can be dependent fields.

E. Checkbox fields can be controlling fields but not dependent fields

ANSWER: A B C E

Explanation:

Field dependencies limit the values users can select in a dependent picklist according to the value selected in its controlling field, improving guided data entry and data quality. A custom picklist field can serve in either role: it can control the available values in another field, or it can be configured as the dependent field whose available values are filtered. A checkbox can be used as a controlling field, allowing administrators to define different dependent values for its selected and unselected states. A multi-select picklist can be configured as a dependent field, although it cannot be used as the controlling field. In addition, data imports must respect configured dependencies: when a record includes a dependent-picklist value, that value must be valid for the controlling-field value on the same record. This prevents imported records from bypassing the dependency's allowed value combinations. These capabilities directly support Universal Containers' goal of linking field choices and presenting users with contextually appropriate values. Note that the prompt says to choose three, but four listed statements accurately describe supported dependency or import behavior. See Salesforce's field dependency documentation and Data Import Wizard documentation.

QUESTION NO: 9

An app builder needs to change the data types of a few custom fields. The app builder is not able to delete and recreate any of the fields, nor modify any Apex code.

What data type change will require the app builder to perform additional steps in order to retain existing functionality?

A. Changing the data type of a field used in lead conversion from number to text.

B. Changing the data type of a field used as an External ID from number to text.

C. Changing the data type of a field used in an Apex class from number to text.

D. Changing the data type of a field used in a report from text to an encrypted field.

ANSWER: A

Explanation:

Changing the data type of a field used in lead conversion from number to text requires additional work because lead conversion field mappings depend on compatible data types between the lead field and the corresponding account, contact, or opportunity field. A number field and a text field cannot remain mapped as the same conversion mapping. Before making the change, the app builder must review the affected lead mappings and ensure that an appropriate compatible destination field exists. After the type conversion, the app builder must update or recreate the field mapping as needed so that values continue to transfer when leads are converted.

This is important because lead conversion is a metadata-driven process: mappings determine which lead values populate the records created during conversion. Preserving the field itself does not by itself preserve a valid mapping when its type no longer matches the mapped target field. The required follow-up is therefore configuration work in Setup, rather than deleting and recreating the custom field or changing Apex. Salesforce documents the requirement to map compatible custom lead fields for conversion in its Lead Conversion documentation and describes field-type considerations in Custom Field Types.

QUESTION NO: 10

What feature can an app builder use to automatically assign cases that have been open longer than three days to the next support tier?

A. Case Assignment Rules

B. Case Escalation Rules

C. Case Business Rules

D. Case Auto Response Rules

ANSWER: B

Explanation:

Case Escalation Rules are designed to automatically take action on cases that remain unresolved beyond a defined period. An app builder can configure an escalation rule with criteria that identifies the relevant cases and an escalation action that runs after three days. The action can reassign the case to a queue or user representing the next support tier, update case fields, send notifications, or perform a combination of these actions. This supports service-level processes by ensuring that cases do not remain with the initial support team after the organization's permitted response or resolution interval has elapsed. Escalation timing is based on the case's age and can be configured using business hours so that the three-day threshold reflects the organization's operating schedule when appropriate. Salesforce evaluates active escalation rules and executes the configured escalation actions when qualifying cases reach the specified age. For implementation details, see Salesforce's Case Escalation Rules documentation and the Escalation Rule Actions documentation.

QUESTION NO: 11

In which of these scenarios is ETL a better choice than Lightning Connect?

A. You need to create or update the external data in addition to reading it.

B. You need the external data to follow the sharing rules defined for your organization.

C. You want to generate reports and charts from the external data.

D. All of the above.

ANSWER: D

Explanation:

All of the above is correct. An ETL process extracts data from an external system, transforms it as needed, and loads it into Salesforce objects. Because the records are stored in Salesforce, they can be created and updated through normal Salesforce data operations, and the organization can apply its standard record-access model, including ownership and sharing rules. Loaded records are also native Salesforce data for reporting purposes, allowing report types, reports, dashboards, and charts to be built using the imported information.

Lightning Connect, now generally associated with Salesforce Connect, is designed to provide access to data that remains in an external system through external objects. That virtualized approach is useful when data should not be copied into Salesforce, but ETL is the stronger architectural choice when the business requires Salesforce-managed data operations, Salesforce sharing behavior, and broad native analytics on the data. The Salesforce integration-pattern guidance describes data migration and replication as appropriate approaches when external data needs to be brought into Salesforce, while Salesforce Connect documentation explains the external-object model. See Salesforce Integration Patterns and Salesforce External Objects overview.

QUESTION NO: 12

What external relationship requires a Salesforce ID in the external data?

A. External Lookup

B. Indirect Lookup

C. Lookup

D. Master Detail

ANSWER: A

Explanation:

External Lookup is correct because this relationship type uses a Salesforce record ID stored in the external data to identify the related parent record. It can link a child external object to a parent external object, or link a standard or custom child object to an external parent object. The relationship field must contain the Salesforce ID of the parent external-object record, allowing Salesforce Connect to resolve the association when the external data is accessed.

This is useful when the external system can store the Salesforce-generated ID for the applicable external-object record. The ID provides a direct, unambiguous reference to the parent record in Salesforce. Salesforce documents that an external lookup relationship identifies the parent using its Salesforce record ID, whereas relationships based on matching external identifiers are configured differently. For implementation details and relationship behavior, see Salesforce Help's External Object Relationships documentation and the Salesforce Connect relationship learning module.

QUESTION NO: 13

DreamHouse Realty (DR) is expanding into subsidized housing by partnering with local government entities. DR uses Sales Cloud and has Enabled field history tracking on the Opportunity object. Due to increased information requirements, the App Dev team is changing Text Area (Long) fields To Rich Text fields to allow for up to 1,000 characters and better descriptions. Which two considerations should be made by the team? Choose 2 answers

A. Data loss may occur when changing custom field types.

B. Audit Trail is available through REST API extracts.

C. Rich text field values of all lengths are displayed fully in reports.

D. Field History Tracking records value changes of 255 characters or less.

ANSWER: A D

Explanation:

Data loss may occur when changing custom field types. Salesforce advises teams to evaluate the impact before changing a custom field's data type, because some conversions can affect existing data, field attributes, dependencies, and downstream uses such as formulas, integrations, reports, and automation. The team should review the conversion in a sandbox, back up affected Opportunity data, and validate that the new rich text field behavior satisfies the business requirement before applying the change in production. Salesforce's field-type documentation is available at Custom Field Types.

Field History Tracking records value changes of 255 characters or less. For long text and rich text content, history tracking can identify that a change occurred, but complete prior and new field values are available only when the tracked value is within the 255-character limit. This is particularly important for subsidized-housing descriptions, where users may enter detailed narrative content and auditors may expect to see the exact before-and-after text. The team should therefore treat field history as evidence of a modification rather than as a complete version history for lengthy rich text entries, and design an appropriate retention or audit approach if full text revisions are required. See Salesforce's guidance at Track Field History for Custom Objects.

QUESTION NO: 14

An app builder has modified a Lightning record page for a case and has added an email button item to the page layout; however, users are unable to see the new item on the layout.

What are two potential reasons why users are unable to view the item on the Case Lightning record page?

Choose 2 answers

A. The page layout includes the case feed component.

B. The email button contains JavaScript.

C. The case page layout also contains custom buttons.

D. The page layout excludes the case feed component.

ANSWER: A B

Explanation:

The page layout includes the case feed component is a potential reason because Case Feed changes how users interact with Case records. Rather than presenting the page layout's standard buttons in the usual location, Case Feed surfaces relevant record actions in the feed-based interface. Therefore, a button added to the traditional page-layout button area might not be displayed where the app builder or users expect on the Lightning Case record page. Salesforce describes Case Feed as a feed-oriented workspace for Cases, with actions integrated into that experience.

The email button contains JavaScript is also a valid reason. JavaScript buttons are not supported in Lightning Experience. Salesforce's Lightning Experience migration guidance identifies JavaScript buttons as unsupported and recommends replacing their behavior with supported alternatives, such as quick actions, Lightning components, Flow, or URL-based functionality where appropriate. Consequently, an email button implemented as a JavaScript button will not be available for users on a Lightning record page. See Salesforce's Case Feed overview and Lightning Experience custom-button migration guidance.

QUESTION NO: 15

AW Computing has a custom object for service plans.

A service plan needs to be associated to one and only one contact. The support manager noticed if the wrong contact is associated, the reps are unable to change the contact. The app builder already confirmed the user has correct access to the field and there are no validations associated with the service

plans.

What could be causing the issue?

A. The Read Only radio button, Allows users with at least Read access to the Master record to create, edit, or delete related Detail records, is selected.

B. The Allow reparenting checkbox, Child records can be reparented to other parent records after they are created, is unchecked.

C. The Read/Write radio button, Allows users with at least Read/Write access to the Master record to create, edit, or delete related Detail records, is selected.

D. The Allow reparenting checkbox, Child records can be reparented to other parent records after they are created, is checked.

ANSWER: B

Explanation:

The Allow reparenting checkbox, Child records can be reparented to other parent records after they are created, is unchecked. A required association to exactly one Contact is consistent with a master-detail relationship in which the Contact is the master record and Service Plan is the detail record. In a master-detail relationship, Salesforce can prevent users from changing a detail record's master after the detail record has been created. This behavior is controlled by the relationship field's *Allow Reparenting* setting rather than by ordinary field-level security or validation rules.

When reparenting is not enabled, the initially selected Contact remains the Service Plan's parent. Consequently, a representative can have permission to view and edit the Service Plan and access to the relationship field, but still cannot replace the existing Contact with another Contact. Enabling reparenting on the master-detail relationship permits moving an existing Service Plan record to a different Contact, subject to the user having the required record access. This setting is specifically intended to govern changes to the parent of an existing detail record.

Salesforce describes the behavior and considerations for master-detail relationships in its relationship considerations documentation and provides relationship-field configuration details in the Master-Detail Relationship reference.

QUESTION NO: 16

When an opportunity closes, close all activities related to that opportunity automatically and create a renewal opportunity.

Which tool would you use for the following use case?

A. Process builder

B. Flow

C. Workflow

D. Approvals

ANSWER: A

Explanation:

Process builder is the appropriate tool for this use case in the context of the Spring '19 Platform App Builder exam. A process can be configured on the Opportunity object and evaluated when the opportunity is created or edited. Its criteria can detect that the opportunity has reached a closed stage, such as Closed Won. Once that condition is met, the process can perform immediate automation actions, including creating a new renewal Opportunity and coordinating the updates needed for the opportunity's related activity records. This makes it suitable for a business event that requires more than a simple field update on the record that triggered the automation.

The automation is declarative, so an app builder can implement the close-and-renew business process without Apex code. The process is also tied directly to the Opportunity lifecycle, ensuring that the renewal is created consistently whenever the defined closing condition occurs. Salesforce's process-automation guidance describes Process Builder as a historical declarative tool for automating actions when records meet specified criteria. Although Salesforce now recommends Flow Builder for new automation, Process Builder was the expected automation tool for this exam-era scenario. See Salesforce's Process Builder overview and choose the right automation tool guidance.

QUESTION NO: 17

Universal Containers needs to retain an audit trail showing the date and time selected fields on a custom Project record were changed, the user who changed them, and the old and new values.

What should an app builder configure?

A. Field History Tracking

B. Setup Audit Trail

C. Feed Tracking

D. Reporting Snapshot

ANSWER: A

Explanation:

Field History Tracking records changes to selected fields on a standard or custom object. The history related list can display the date and time of the change, the user who made the change, and the prior and new field values.

After enabling history tracking for the Project object, the app builder selects the fields to track and adds the Project History related list to the page layout. This provides a declarative audit trail without creating custom automation. See Salesforce guidance on Field History Tracking.

QUESTION NO: 18

What is a self-relationship?

A. A lookup field to the user object.

B. A lookup to the global search.

C. A relationship of account to opportunity owned by the same owner.

D. A lookup field to the same object.

ANSWER: D

Explanation:

A self-relationship is correctly defined as “A lookup field to the same object.” In Salesforce, this is created when a lookup relationship field on an object references records of that very same object. It lets records form a parent-child hierarchy without needing a separate parent object. For example, the standard Account object includes the Parent Account field, which is a lookup to another Account record. A company can therefore represent an organizational structure in which a subsidiary account points to a parent account, and that parent can itself have another parent.

Self-relationships are useful whenever records naturally relate to other records of the same type, such as organizational units, product-component structures, or account hierarchies. The lookup stores the related record’s identifier while preserving the flexibility of a lookup relationship: the related record can generally be changed or cleared, subject to field settings and validation rules. Salesforce describes a self relationship as a lookup relationship from an object to itself and uses Account’s Parent Account relationship as the standard example. See Salesforce’s Self Relationships documentation and the Account Hierarchy documentation for the practical implementation of this pattern.

QUESTION NO: 19

What is true about unmanaged packages?

A. Once the components are installed from an unmanaged package, the components can be edited in the organization they are installed in.

B. The developer who created and uploaded the unmanaged package has no control over the installed components, and can't change or upgrade them.

C. Unmanaged packages should not be used to migrate components from a sandbox to production organization. Instead, use Change Sets.

D. All of the above.

ANSWER: D

Explanation:

All of the above is correct because unmanaged packages are intended primarily as a way to distribute open-source components or provide a one-time snapshot of metadata. After installation, the receiving organization owns the installed components and can customize them, including editing supported metadata such as custom objects, fields, flows, and other package contents. The package creator does not retain a managed relationship with those installed components and cannot push upgrades or otherwise control the subscriber's modified copy.

Because an unmanaged package is a one-time distribution mechanism rather than an upgradeable deployment artifact, it is not the recommended approach for promoting configuration between related sandbox and production environments. Change sets are designed for moving supported metadata between organizations that have a deployment connection, such as a sandbox and its associated production org. They provide an explicit outbound and inbound deployment process appropriate for that migration scenario. Salesforce documentation describes unmanaged packages as editable after installation and notes that they cannot be upgraded; it also documents change sets for deploying customizations between connected organizations. See Salesforce Packaging Guide and Salesforce Change Sets documentation.

QUESTION NO: 20

Universal Containers (UC) wants to delete data in several fields for 5,000 Lead records. UC exported the selected record Ids and fields that need to have data deleted in a CSV file. Which two steps should an app builder suggest to meet these requirements? Choose 2 answers

A. Select the correct record type.

B. Use Import Wizard to update leads using the CSV file.

C. Use Data Loader to update leads using the CSV file.

D. Select Insert Null Values in Settings.

ANSWER: C D

Explanation:

Use Data Loader to update leads using the CSV file and select Insert Null Values in Settings. Data Loader is appropriate for a bulk update involving 5,000 existing Lead records because the exported CSV can retain each Lead's Salesforce record ID while identifying the fields whose values must be cleared. Before running the update, the values in the applicable field columns should be blank in the CSV. The update operation then matches each row to the existing Lead through its ID and applies the requested field changes.

In Data Loader, blank cells are not automatically interpreted as instructions to erase field data. Enabling *Insert Null Values* directs Data Loader to process blank values as nulls during insert and update operations. Therefore, when the update file contains Lead IDs and blank cells for the fields to clear, this setting causes Salesforce to remove the existing values in those editable fields. This pairing provides a controlled, repeatable bulk process and supports reviewing the CSV and Data Loader success/error files after execution. Salesforce documents the Insert Null Values setting in the Data Loader configuration parameters and describes using Data Loader for bulk record updates in its Data Loader guide.

QUESTION NO: 21

Which three are features of the Custom Button? (Choose three.)

A. Custom Button with Javascripts enhance Lightning Experience.

B. Custom Button is available for User Object.

C. Custom Button display at the top and bottom of a page.

D. Custom Button is available for Person Account.

E. Custom Button can reference an external app.

ANSWER: B C E

Explanation:

Custom buttons can be created for the User object, allowing an administrator to provide users with a tailored action from the User record context. As with buttons on other supported objects, the button's behavior and placement are configured through the object's button, link, and action settings and the relevant page layout.

A detail-page custom button can be placed in the standard button areas at the top and bottom of a record's detail page. This placement makes the action available while a user is viewing an individual record, rather than requiring the user to navigate elsewhere. Page-layout configuration controls whether an available button is exposed to users.

Custom buttons can also use a URL to direct the user to an external application or resource. URL parameters can incorporate Salesforce merge fields, enabling the destination to receive relevant record information and support integrations or context-sensitive navigation. Salesforce documents custom buttons and links, including URL-based behavior and their use on page layouts, at Custom Buttons and Links. Additional implementation guidance for Salesforce pages and custom button behavior is available in the Salesforce Developers documentation.

QUESTION NO: 22 - (DRAG DROP)

DRAG DROP

In what order does Salesforce process rules?

Match the rules from the left column with their appropriate order in the right column.

Select and Place:

RULES		ORDER
Assignment rules	1	
Auto-response rules	2	
Workflow rules (with immediate actions)	3	
Escalation rules	4	
Validation rules	5	

ANSWER:

RULES		ORDER
Assignment rules	1	Validation rules
Auto-response rules	2	Assignment rules
Workflow rules (with immediate actions)	3	Auto-response rules
Escalation rules	4	Workflow rules (with immediate actions)
Validation rules	5	Escalation rules

Explanation:

The displayed sequence correctly reflects the relevant portion of Salesforce's record save order of execution. Validation rules occur first among the listed rule types because Salesforce must confirm that the record satisfies the organization's data-quality requirements before it continues processing the save. Once those checks pass and the record proceeds through the save process, Salesforce applies assignment rules. Assignment rules are used for Lead and Case records to determine the appropriate owner or queue based on the configured criteria.

Salesforce then processes auto-response rules. These rules can send the appropriate automatic email response for qualifying Lead or Case submissions, using the record information and the outcome of assignment processing. Workflow rules with immediate actions follow. When workflow criteria are met, immediate workflow actions are evaluated and executed as part of the same transaction. Such actions can include field updates, email alerts, tasks, and outbound messages where those legacy workflow capabilities remain in use.

Escalation rules are last in this set. They apply to Case records and establish escalation timing and handling based on the saved case's values, including its ownership and other data finalized during the preceding processing stages. Therefore, the correct mapping is: first Validation rules; second Assignment rules; third Auto-response rules; fourth Workflow rules (with immediate actions); and fifth Escalation rules. Salesforce documents the broader save lifecycle in its Order of Execution reference. Details of workflow evaluation and immediate actions are also covered in the Salesforce Workflow documentation.

QUESTION NO: 23

Ursa Major Solar wants to see the Type field from the parent object Galaxy listed on the child record Star. The app builder is receiving an error stating " Picklist values are only supported in certain functions " . Which formula should an app builder use to achieve the desired result?

A. TEXT(Galaxy__r.Type__c)

B. VALUE(Galaxy__r.Type__c)

C. Galaxy__r.Type__c

D. FIND(Galaxy__r.Type__c)

ANSWER: A

Explanation:

TEXT(Galaxy__r.Type__c) is the appropriate formula because `Type__c` is a picklist field on the related parent Galaxy record and the desired outcome is to display that selected picklist value on the Star record. The `Galaxy__r` portion is the custom relationship reference used in a cross-object formula to traverse from the child record to its parent. Appending `.Type__c` accesses the parent's Type field.

Salesforce formulas require picklist values to be used only through supported picklist functions. `TEXT()` converts the selected picklist value into its text representation, allowing the formula to return and display it in a text-based formula field. For example, if the Galaxy record's Type value is "Spiral," the formula returns "Spiral" on the related Star record. This approach remains dynamic: when the parent Galaxy's Type value changes, the formula result on related Star records reflects the current parent value without copying or maintaining duplicate data.

Salesforce documents `TEXT()` as the function for converting values, including picklists in supported formula contexts, to text. See Salesforce Formula Operators and Functions: TEXT and Salesforce Formula Field Functions.

QUESTION NO: 24

An app builder is loading data into Salesforce. To link the new records back to the legacy system, a field will be used to track the legacy ID on the Account object. For future data loads this ID will be used when upsetting records. Which two field attributes should be selected? Choose 2 answers

A. Unique

B. Text (encrypted)

C. Required

D. External ID

ANSWER: A D

Explanation:

Unique and **External ID** should be selected for the legacy-ID field. An external ID is a custom field that stores an identifier from a system outside Salesforce. Marking the field as an external ID enables Salesforce data tools and APIs to use the legacy value as the matching key when performing an upsert. During an upsert, Salesforce locates an existing Account with the supplied external-ID value and updates it; if no matching Account is found, Salesforce creates a new record. This supports repeatable migration and integration processes without requiring Salesforce record IDs to be maintained in the legacy application.

The field should also be unique so every legacy-system identifier maps to only one Account. A one-to-one match is essential for dependable upsert processing: it prevents more than one Account from being associated with the same legacy record and makes the matching result unambiguous over future imports. Together, these attributes establish the legacy ID as a durable integration key, allowing the app builder to load data initially and safely synchronize subsequent updates. Salesforce documents external-ID upsert behavior in the REST API Upsert documentation and describes external IDs among supported Salesforce field types.

QUESTION NO: 25

Which three are true about converting a tabular, summary, or matrix report to a joined report? (Choose three.)

A. Cross filters are not supported in joined reports.

B. The Rows to Display filter is not supported in joined reports.

C. Joined report blocks are formatted as matrix reports.

D. Report formula fields are not supported in joined reports.

E. Bucket fields are not supported in joined reports.

ANSWER: A B E

Explanation:

When a standard report is converted to a joined report, Salesforce applies joined-report feature limitations because the report consists of separate report blocks that can use different report types and filters. Consequently, cross filters are not supported in joined reports. Cross filters ordinarily refine a report based on the presence or absence of related records, but they cannot be retained or added in the joined-report format. The Rows to Display filter is also not supported. This filter is the report row-limit capability used to restrict a tabular report to a specified number of returned rows, and joined reports do not support that limitation. Bucket fields are likewise unavailable in joined reports. Bucketing categorizes report values into user-defined groups, but it is not a supported feature when working with joined report blocks. Therefore, these unsupported features must be removed or replaced with supported filters, groupings, or formulas as appropriate before or after conversion. Salesforce's joined-report guidance and report limitations document the supported configuration model for joined reports and their blocks. See Salesforce Help: Joined Reports and Salesforce Help: Report Formats.

QUESTION NO: 26

Universal Containers is rolling out a new opportunity review process. Regional managers will need to edit opportunities for their subordinates, but not for other groups. Managers and users should be able to view all opportunities. Which two approaches are recommended to meet their requirement? (Choose two.)

A. Set organization-wide defaults to public read/only.

B. Create standard role hierarchies

C. Set organization-wide defaults to public read/write.

D. Create criteria based sharing rules.

ANSWER: A B

Explanation:

Set organization-wide defaults to public read/only and create standard role hierarchies together provide the required baseline visibility and selective edit access. With Opportunity organization-wide defaults set to Public Read Only, every user can view Opportunity records across the organization, satisfying the requirement that both managers and users can view all opportunities. Record owners and users granted additional access can still edit as appropriate.

A standard role hierarchy models the management structure, placing regional managers above their subordinate users. For standard objects such as Opportunity, access granted through the role hierarchy lets users higher in the hierarchy access records owned by users beneath them. This enables each regional manager to edit opportunities owned by their subordinates while preserving the intended separation between unrelated regional groups. The combination applies least-privilege sharing: broad read access is available to everyone, while write access is inherited only through the relevant reporting structure. Salesforce documents organization-wide defaults and role-hierarchy access in its record-sharing overview and explains how role hierarchies provide record access to users above record owners.

QUESTION NO: 27

Universal Containers wants to give sales managers the ability to quickly provide sign off On an Opportunity vi a the Opportunity record page when a sales rep has discounted a deal by 20% To 30%. Which two features should be used for this requirement? Choose 2 answers

A. Dynamic Actions

B. Schema Builde

C. Approval Process

D. Validation Rule

ANSWER: A C

Explanation:

Dynamic Actions and **Approval Process** together meet the requirement for conditional, manager-controlled sign-off from the Opportunity record page. An approval process supplies the formal approval workflow: it can use Opportunity entry criteria that identifies discounts in the 20% through 30% range, route the submitted Opportunity to the appropriate sales manager, and record the manager's approval or rejection as part of the record's approval history. This creates an auditable sign-off process rather than merely displaying information or preventing data entry.

Dynamic Actions make the relevant record-page action readily available in Lightning Experience. An app builder can configure action visibility using filters so that the approval-related action, such as submitting the record for approval, is shown when the Opportunity's discount falls within the stated range. Sales users can therefore initiate the required process directly from the Opportunity page at the appropriate time, while managers receive and complete the formal approval request through the configured approval process.

Salesforce documents approval routing, criteria, and actions in its Approval Processes overview. Conditional action placement on Lightning record pages is covered in the Dynamic Actions overview.

QUESTION NO: 28

What is true when a field update is set to re-evaluate the workflow rule? (Choose three.)

A. In a batch update, workflow is only re-triggered on the entities where there is a change.

B. Any workflow rules whose criteria are met as a result of the field update will be ignored.

C. Only workflow rules on the same object as the initial field update will be re-evaluated and triggered.

D. Cascade of workflow rule re-evaluation and triggering can happen up to ten times after the initial field update that started it.

E. Only workflow rules that didn't fire before will be re-triggered.

ANSWER: A C E

Explanation:

When a workflow field update is configured to re-evaluate workflow rules after the field change, Salesforce performs another evaluation only when the update actually changes the record's data. Consequently, during a batch operation, reevaluation applies to the individual records for which the field update produced a change. This avoids unnecessary workflow processing for records whose stored values remain the same.

The reevaluation is limited to workflow rules defined on the object whose record was updated. Salesforce reevaluates those rules against the record's new values, enabling a rule whose criteria have newly become true to execute its associated actions. This behavior supports controlled workflow chaining on a single object while maintaining predictable transaction processing.

Salesforce also prevents a workflow rule from firing repeatedly in the same transaction after it has already fired. Therefore, reevaluation can trigger workflow rules that have not previously fired, allowing additional qualifying automation to run without uncontrolled repetition. These mechanics are documented in Salesforce's workflow field-update guidance and workflow-rule considerations: Re-evaluate Workflow Rules After Field Change and Workflow Rules Considerations.

QUESTION NO: 29

Northern Trail Outfitters wants to change a master-detail Relationship on Account to a lookup relationship with a custom object Park. The app builder tries to reconfigure this but is unable to do so. What could be causing this?

A. The Account is included in a flow process on the Park object.

B. The park records have existing formulas on the Account.

C. The Park object needs at least one Master-Detail field for reporting.

D. The Account record includes Parks roll-up summary fields.

ANSWER: D

Explanation:

The Account record includes Parks roll-up summary fields is correct because roll-up summary fields depend on a master-detail relationship. A roll-up summary field is created on the master object and calculates values from its related detail records, such as a count of Park records or a sum of a numeric field on those records. In this scenario, Account is the master and Park is the detail object, so any Account roll-up summary that summarizes Park data relies on the existing master-detail relationship.

Salesforce does not allow that master-detail relationship to be changed to a lookup relationship while dependent roll-up summary fields exist. Before converting the relationship, the administrator must identify and remove the applicable roll-up summary fields from Account. If the summarized information is still required after conversion, it can be replaced using an appropriate alternative, such as a record-triggered flow, depending on the business requirement. Salesforce's relationship-field guidance describes the conversion limitation, and its roll-up summary documentation explains that roll-ups summarize records across master-detail relationships.

See Salesforce relationship considerations and Salesforce roll-up summary fields.

QUESTION NO: 30

A salesperson at AW Computing only sees the Social Contacts link for Twitter and not Facebook on his records.

Why would this be happening?

A. None of his Facebook contacts have confirmed the nature of their relationship.

B. The administrator hasn't enabled Social Contacts for Facebook.

C. The salespersons login with Facebook has expired.

D. Facebook is no longer supported by Social Contacts.

ANSWER: B

Explanation:

The administrator hasn't enabled Social Contacts for Facebook. Social Contacts integrations are configured by social network, so enabling and authorizing Twitter does not automatically make Facebook available. An administrator must complete the Facebook-specific setup and enable that provider before Salesforce can display Facebook social-contact functionality on supported person records. Once that provider-level configuration is complete, Salesforce can use the Facebook connection when presenting Social Contacts information, subject to the organization's feature availability and the user's applicable access. This separate-network setup model lets an organization choose which external social services it connects to and ensure each connection is authorized appropriately. Salesforce's Social Contacts implementation documentation describes the required administrative setup for supported social-network integrations, including configuring individual providers: Salesforce Social Contacts Implementation Guide. Salesforce also maintains current setup and feature documentation in Salesforce Help, where administrators can verify availability and configuration requirements for their release.

QUESTION NO: 31

If ABC Company wants to refresh a sandbox weekly with some example data, which sandbox should ABC Company create?

A. Enterprise Sandbox

B. Developer Pro Sandbox

C. Partial Copy Sandbox

D. Full Sandbox

E. Developer Sandbox

ANSWER: C

Explanation:

Partial Copy Sandbox is the appropriate choice because it is designed for development, testing, and training scenarios that need a representative subset of production data rather than an entire production-data copy. When creating or refreshing this sandbox type, an administrator can use a sandbox template to select specific objects and define the sample records to include. This lets ABC Company provide useful example data while keeping the sandbox smaller and faster to manage than a full copy. A Partial Copy Sandbox has a five-day refresh interval, so it can be refreshed weekly as requested. The weekly cadence also allows the organization to keep its sample dataset reasonably current for ongoing testing without requiring a full production refresh. Salesforce documents that Partial Copy sandboxes include metadata plus a selected sample of production data, with a five-day refresh interval. Review the sandbox types and refresh intervals in Salesforce Help: Sandbox Types and Refresh Intervals. For details on selecting the data included during refreshes, see Sandbox Templates.

QUESTION NO: 32

7 of 65. Universal Container ' s sales reps can modify fields on an opportunity until it Is close The sales operations team has access to modify the Post-Close Follow-up Date and Post-Close Follow-up Comments fields after the Opportunity is close After the opportunity is closed, the rest of the fields Are read only. How should these requirements be met?

A. Use field-level security on page layouts to restrict editing fields.

B. Use field-level security on page layouts with record types to restrict editing fields.

C. Use record types with field sets and restrict editing fields using field-level security.

D. Use field-level security to mark fields as read-only on the Sales profile.

E. Use a validation rule to restrict post-close edits, allowing only the designated follow-up fields for authorized sales operations users.

ANSWER: E

Explanation:

Use a validation rule to enforce the edit restrictions based on whether the Opportunity is closed and whether the user is authorized to perform post-close updates. The rule can evaluate the Opportunity's closed status with `IsClosed`, identify changes by comparing fields with `ISCHANGED()`, and allow only changes to Post-Close Follow-up Date and Post-Close Follow-up Comments for members of the sales operations team. A custom permission assigned through a permission set is a maintainable way to identify authorized sales operations users, rather than hard-coding profile names in the formula. The validation rule should prevent edits to all other fields after close, while allowing the designated users to update the two approved follow-up fields. It can also prevent non-authorized users from changing either post-close field after the Opportunity has closed. Validation rules run whenever a record is saved, so the restriction is consistently enforced regardless of whether the edit originates in the standard Salesforce interface, a mobile client, an integration, or an API-based process. Salesforce documents validation rules as formula-based checks that block saves and display an error when data fails business requirements. See Salesforce Validation Rules and Salesforce Custom Permissions.

QUESTION NO: 33

An app builder wants to limit the number of fields users are required to fill out when creating a new Opportunity. Once they fill out the required fields and save, the full record page with additional fields relevant to the Opportunity type becomes available.

How could this be accomplished?

A. Make the Opportunity type a required field on the initial Opportunity page layout and use automation to fill in the type field to a record type.

B. Use different page layouts for Opportunity types based on the user profile.

C. Once the required fields are populated, use a sharing rule to share the new fields with the user.

D. Hide additional sections on the page layout and show the users how to manually expand them when they want to fill in the fields in the hidden sections.

ANSWER: A

Explanation:

Make the Opportunity type a required field on the initial Opportunity page layout and use automation to fill in the type field to a record type. This approach uses record types and their associated page layouts to provide a streamlined initial creation experience followed by a more complete, context-specific record experience. The initial layout can contain only the essential fields, including a required Opportunity Type value. After the record is saved, automation can use that value to assign the appropriate record type. Salesforce then presents the page layout assigned to that record type, making the additional fields that are relevant to the selected kind of Opportunity available to the user.

Record types are designed to support distinct business processes and page layouts for the same object. A record type can therefore represent each Opportunity category, while its assigned layout controls which fields and sections users see after creation. This keeps the initial data-entry burden low without losing the ability to capture detailed information later in the process. See Salesforce Trailhead's overview of record types and page layouts and Salesforce Help on workflow field updates.

QUESTION NO: 34

The organization-wide default for a custom object is set to private. The Supervisor profile grants view access to the same object. A user with the Supervisor profile is also listed as the Manager on the user detail records for a subordinate. However, the Supervisor still cannot view records owned by the subordinate.

Which two issues are preventing the Supervisor from viewing records owned by the subordinate? (Choose two.)

A. Organization-wide settings for the custom object grant access to other users with the same role.

B. The Supervisor requires a permission set in order to view the subordinate's records.

C. The Supervisor's role is not above the subordinate's role in the hierarchy.

D. Organization-wide settings for the custom object do not grant access using hierarchy.

ANSWER: C D

Explanation:

"The Supervisor's role is not above the subordinate's role in the hierarchy" is an issue because the role hierarchy—not the Manager field on a user record—provides implicit record access based on reporting structure. With private organization-wide defaults, a user above a record owner in the role hierarchy can ordinarily gain access to that owner's records through the hierarchy. Simply identifying a Supervisor as someone's manager does not establish this sharing relationship; the users must be placed in appropriate roles, with the Supervisor's role higher than the subordinate's role.

"Organization-wide settings for the custom object do not grant access using hierarchy" is also an issue when the custom object's *Grant Access Using Hierarchies* setting is disabled. For custom objects, administrators can control whether users higher in the role hierarchy automatically receive access to records owned by users below them. This setting must allow hierarchy access, in addition to having the roles arranged correctly, for hierarchy-based access to apply to private custom-object records. Salesforce describes role hierarchy access in its Roles and the Role Hierarchy module and documents the configurable behavior for custom-object sharing in Sharing Rules.

QUESTION NO: 35

Global actions can be created to let users create which of the following kinds of records? (Choose three.)

A. Question

B. Event (without invitees)

C. Products

D. Opportunity

E. Chatter Post

F. Users

ANSWER: A B D

Explanation:

Global actions are configured in Setup and added to a global publisher layout so that users can initiate common work from places that are not tied to a specific record. **Question** is supported through the Chatter question publisher capability, allowing users to ask a question from the global action menu. **Event (without invitees)** is supported as a global create action: users can create the Event record, but global Event actions do not support adding invitees. This makes global actions useful for quickly logging an individual calendar event while avoiding record-specific context. **Opportunity** is also a supported standard-object target for a global create action. Because the action is global rather than object-specific, Salesforce does not automatically populate a relationship to a particular parent record; users provide the required Opportunity information in the action layout. Administrators can tailor the action layout and place these actions in the publisher according to the organization's workflow. Salesforce's action overview and Trailhead guidance describe how global actions support creating records and publishing actions independently of a current record page. See Salesforce Help: Quick Actions Overview and Trailhead: Create Quick Actions.

QUESTION NO: 36

Cloud Kicks works on an annual subscription model. When a sales rep marks An opportunity as closed won, a new opportunity should automatically be created for The renewal. The contracts team works outside of Salesforce but also needs to be Notified about closed deals in order to initiate the contract process with the customer. Which automation solution would meet these requirements?

A. Approval Process

B. Outbound Message

C. Record-triggered flow

D. Validation Rule

ANSWER: C

Explanation:

Record-triggered flow is the appropriate solution because it can run automatically when an Opportunity is updated and meets defined entry conditions, such as when its stage changes to Closed Won. The flow can then use a Create Records element to generate a new renewal Opportunity, copying or setting the relevant account, close date, amount, and renewal-related values as required by the business process. This provides the needed record creation without requiring a user to initiate a separate action.

The same flow can notify the contracts team as part of the automated process. For example, it can use an email alert or a Send Email action to send a notification to recipients who work outside Salesforce, provided their email addresses are available through the selected notification configuration. The automation is therefore able to coordinate both post-sale activities from the single Closed Won event: creation of the renewal pipeline record and communication that allows the contracts process to begin.

Salesforce documents that record-triggered flows run when records are created or updated and can perform actions such as creating related records and sending email notifications. See Salesforce Record-Triggered Flows and Trailhead: Build Logic in a Flow.

QUESTION NO: 37

Which of the following are types of developer sandboxes environment types in Salesforce? (Choose four.)

A. Developer

B. Developer Pro

C. Partial Copy

D. Full Sandbox

E. Partial Sandbox

F. Full Copy

ANSWER: A B C D

Explanation:

Salesforce provides four standard sandbox types: Developer, Developer Pro, Partial Copy, and Full. These environments are separate copies of a production org used to safely configure, develop, test, train, and validate changes without affecting production data or users. Developer sandboxes are intended for coding and configuration work and include a small data and file-storage allocation. Developer Pro sandboxes serve similar development purposes but provide a larger storage allocation, making them suitable for more substantial development and testing needs. Partial Copy sandboxes include selected production data defined through a sandbox template, so they support testing that requires representative records. Full sandboxes are complete copies of production, including all data and metadata, and are used for activities such as full

integration testing, user acceptance testing, and performance testing. Salesforce identifies these exact names as the available sandbox types; therefore, the selections use the official product terminology, including “Partial Copy” and “Full.” For details on each sandbox’s purpose, data copy behavior, and refresh interval, see Salesforce Help: Sandboxes and Salesforce Developer Guide: Sandboxes.

QUESTION NO: 38

Universal Containers uses a private Account sharing model. They have a Process Improvement team with representatives from multiple departments that needs to view all accounts that have been flagged as problem accounts.

How should this team be granted access to the records?

A. Use a record owner sharing rule that is shared with the Process Improvement public group.

B. Use a criteria-based sharing rule where the accounts are shared with the Process Improvement public group.

C. Write a trigger to use Apex Managed Sharing to grant access with the Process Improvement team.

D. Use a record owner sharing rule that is shared with the Process Improvement role.

ANSWER: B

Explanation:

Use a criteria-based sharing rule where the accounts are shared with the Process Improvement public group. With Account organization-wide defaults set to Private, users can access only accounts they own or that have been explicitly shared with them. A criteria-based sharing rule is designed for exactly this requirement: it automatically shares records that meet defined field criteria, such as an Account field indicating that the account is a problem account.

The Process Improvement team includes people from multiple departments, so a public group provides a flexible way to collect those users in one share target without depending on their roles or reporting hierarchy. The rule can grant the public group Read access to every Account whose problem-account flag meets the specified condition. It also remains dynamic: when an Account is flagged, Salesforce applies the sharing rule; if it no longer meets the criteria, access granted by that rule is removed. This declarative approach is maintainable and uses Salesforce’s standard record-sharing model. See Salesforce Sharing Rules documentation and Salesforce Public Groups documentation.

QUESTION NO: 39

Cloud Kicks has five years of sales data and would like to track when customers made their first purchase. How should an app builder use a roll-up summary to meet the requirements?

A. Create a new date field called First Order Date, then create roll-up summary to update the field using Type MIN.

B. Create a new roll-up summary field called First Order Date, Using type MIN on the Opportunity Close Date with a filter where iswon = TRUE.

C. Create a new date field called First Order Date, then set the date using a roll-up summary on Opportunity Close Date.

D. Create a new roll-up summary field called First Order Date using type SUM on Opportunity Close Date.

ANSWER: B

Explanation:

Create a new roll-up summary field called First Order Date, Using type MIN on the Opportunity Close Date with a filter where iswon = TRUE. This correctly derives the date of each customer’s first completed purchase from historical Opportunity records. The roll-up summary belongs on the Account, where it can evaluate related Opportunities and return one value for each customer. Choosing the MIN summary type returns the earliest, or lowest, Close Date among the records included in the roll-up. Because Close Date is a date field, it is an appropriate field for a MIN calculation.

The filter for won Opportunities is essential because a customer's first purchase should be based on a successfully closed sale, rather than on open, lost, or otherwise unsuccessful sales opportunities. Using the IsWon field set to true limits the calculation to records that represent completed revenue. As new Opportunities are marked won in the future, Salesforce automatically recalculates the roll-up and retains the earliest qualifying Close Date, so the field continues to represent the customer's first purchase without automation or manual updates. Salesforce describes roll-up summary fields and supported summary operations in its Roll-Up Summary Fields Trailhead unit. The standard Opportunity fields, including Close Date and IsWon, are documented in the Opportunity object reference.

QUESTION NO: 40

Universal Containers requires e-mails to be sent to additional recipients when a workflow e-mail alert is triggered from the case object.

Which two field types need to be added to the case object to allow additional recipients on the e-mail alert? (Choose two.)

A. Text field

B. E-mail field

C. Lookup field

D. Formula field

ANSWER: B C

Explanation:

An **E-mail field** on Case is appropriate when Universal Containers needs to store an additional recipient's email address directly on each case. Salesforce email alerts can use an email-address field on the record as a recipient source, allowing the alert to be sent to the address entered in that field. This supports recipients who are not necessarily Salesforce users or related contacts.

A **Lookup field** is also appropriate when the additional recipient should be represented by a related Salesforce record, such as a User. A lookup relationship on Case provides a record-based recipient association, which can be selected when configuring the email alert's recipients. This is useful when the recipient should be managed as a person in Salesforce rather than as a manually entered address, and it lets the alert use that related person's email address.

Together, these field types support the two standard patterns for adding dynamic email-alert recipients: storing a direct email address on the case and relating the case to a recipient record. Salesforce documents email alert recipients and the use of email fields in workflow and approval email alerts in Create Email Alerts and describes lookup relationships in Object Relationships Overview.

QUESTION NO: 41

A user is unable to use inline editing on a list view. A quick check verifies the user should be able to perform inline editing as they have been Assigned the appropriate permissions. Which condition should the app builder review?

A. If the list view selected is locked by another user

B. If the list view restricts sharing for the user

C. If the list view contains a chart created by the user

D. If the list view contains more than one record type

ANSWER: D

Explanation:

If the list view contains more than one record type is the condition to review. Inline editing in Salesforce list views has record-type-specific behavior: when the displayed records span multiple record types, Salesforce does not support editing

those records directly from the list view. Even when the user has the object permissions, field-level access, and record access needed to edit, the composition of the list itself can prevent inline editing.

The app builder should inspect the list view's filters and determine whether its results include records assigned to different record types. If so, create or adjust a filter so that the view returns records from one record type only. A record type filter can make the list suitable for inline editing while also presenting users with a more consistent set of fields, business processes, and picklist values. This is particularly relevant for objects that use multiple sales processes, support processes, or business-specific layouts.

Salesforce documents the limitations and supported behavior for inline editing in its list-view guidance. See Inline Editing in List Views and Create and Customize List Views.

QUESTION NO: 42

Which two are true statements about record types? (Choose two.)

A. They can be used to control user role hierarchy.

B. They allow different page layouts and mandatory fields.

C. They can be enabled by profile and permission set.

D. They allow different picklist values for all picklist fields.

ANSWER: B C

Explanation:

Record types allow an organization to present different business processes and user experiences for the same object. They can be associated with different page layouts, so users assigned a particular record type can see fields, sections, buttons, and related lists tailored to that process. Page-layout configuration can also make fields required for users working with that layout, supporting different mandatory data-entry needs without changing the object-level field definition for every user.

Record types can also be assigned through both profiles and permission sets. A profile can specify which record types are available for an object and which one is the default. Permission sets can extend record-type access to selected users without requiring a profile change, which is useful when access needs to be granted to a subset of users or managed independently from their baseline profile. Together, page-layout assignment and profile or permission-set record-type availability let an app builder support distinct processes while maintaining a single underlying object and data model. Salesforce documents these capabilities in its Record Types overview and record type assignment documentation.

QUESTION NO: 43

How can you make a field mandatory? (Choose three.)

A. Create a Workflow Rule.

B. Check the required field box on the page layout.

C. Check the required field box on the field definition.

D. Creating a validation rule for the field.

E. Check the required field box for the field on the record type.

ANSWER: B C D

Explanation:

A field can be made mandatory at several layers, depending on where and how records are created or updated. Checking the required field box on the page layout makes users supply a value when they create or edit a record through that page

layout in the Salesforce user interface. This is useful when the requirement is specific to a particular layout or user experience.

Checking the required field box on the field definition creates a field-level requirement. For supported field types, Salesforce requires a value whenever a record is saved, including records created through the user interface, imports, integrations, and API operations. This is the strongest declarative choice when the field must always contain a value.

Creating a validation rule for the field also makes the field mandatory by preventing a save when specified conditions are met and the field is blank. Validation rules are especially valuable when the requirement is conditional, such as requiring a field only for a certain record type, status, or business process. Salesforce evaluates the rule before committing the record and displays the configured error message if the requirement is not met. See Salesforce guidance on validation rules and page layouts.

QUESTION NO: 44

Service Agents are required to confirm a user ' s identity before providing support information over the phone. Which feature can an app builder use to help agents meet this requirement?

A. Add Path to the top of the Case layout

B. Case Validation Rules

C. Include Surveys as a Case related list

D. Guided Action Flows on the record page

ANSWER: D

Explanation:

Guided Action Flows on the record page is the appropriate feature because it can present service agents with a structured, repeatable set of steps while they work a Case. An app builder can configure guided actions so that an agent selects an identity-verification task from the Case record and is launched into a screen flow. The flow can prompt the agent to ask the required verification questions, capture the responses, apply decision logic, and direct the agent to the next step only after the required checks are complete. This makes the verification process consistent across agents and embeds it directly in the service workflow rather than relying on informal scripts or manual reminders.

Guided actions are designed to surface recommended tasks and flows in the context of a record, helping representatives complete business processes in a prescribed order. For this requirement, the identity-confirmation flow can be available as a task before agents disclose protected support details. The flow can also update the Case with verification status or related information when appropriate, creating an auditable part of the support process. See Salesforce's Flow for Service Trailhead module and the Salesforce Flow documentation for details on using flows to guide users through business processes.

QUESTION NO: 45

An app builder is creating a custom Lightning app for the Sales Operations team.

Which two items can be configured as part of a custom Lightning app? (Choose two.)

A. Navigation items

B. Role hierarchy settings

C. Utility bar

D. Page layout assignments

ANSWER: A C

Explanation:

Navigation items can be configured as part of a custom Lightning app. The app builder can choose the objects, tabs, Lightning pages, and other supported items that users can access from the app navigation bar.

A utility bar can also be configured for a custom Lightning app. Utility bar items provide tools that users can access from anywhere within the app, such as Notes, Open CTI Softphone, or a custom utility component. Page layouts and role hierarchy settings are configured separately and are not components of a Lightning app definition. See Salesforce documentation on Create and Customize Lightning Apps.

QUESTION NO: 46

When an opportunity close date is delayed by more than 60 days, the manager and the VP of Sales must approve the change.

Which two solutions will meet the requirement? (Choose two.)

A. Build an approval process that requires unanimous approval from the manager and VP of Sales.

B. Build a validation rule that does not allow a user to save the opportunity record.

C. Create a workflow rule that checks for close date less than 60 days and add an e-mail alert.

D. Create a Process Builder flow that submits the record for an approval process.

ANSWER: A D

Explanation:

Build an approval process that requires unanimous approval from the manager and VP of Sales is appropriate because an approval process can route an Opportunity through multiple approval steps and require each designated approver to approve before the request is fully approved. The process can use entry criteria that identify the relevant delayed-close-date changes, then assign approval steps to the Opportunity owner's manager and the VP of Sales. Salesforce approval processes also provide the controlled approval status and record-locking behavior typically needed while a change awaits authorization.

Create a Process Builder flow that submits the record for an approval process is also appropriate because automation can submit an Opportunity for approval when the close-date-change condition is met. The process evaluates the record update and invokes the existing approval process without requiring the user to manually click Submit for Approval. Together, automated submission and a multi-step unanimous approval process ensure that qualifying date delays are routed consistently to both required approvers. Salesforce documents approval processes and their sequential approval steps in its Approval Processes overview, and documents automated approval submission through process automation in Submit for Approval actions.

QUESTION NO: 47

Which of these is not a valid report type?

A. Detailed

B. Summary

C. Matrix

D. Tabular

ANSWER: A

Explanation:

Detailed is the correct answer because it is not a Salesforce report format (the terminology intended by "report type" in this question). Salesforce reports are presented in supported formats such as tabular, summary, and matrix. A tabular report displays records in rows, while summary and matrix reports organize data into grouped structures for aggregation and

analysis. Salesforce documentation and learning materials identify these supported report formats; “Detailed” is not included as a selectable report format. Although reports can certainly display detailed record-level rows, that capability is a characteristic of report results rather than the name of a report format. Therefore, “Detailed” does not identify a valid Salesforce report type/format in this context. See Salesforce Trailhead’s discussion of creating and formatting reports at Create Reports and the Salesforce help documentation on report formats at Report Formats.

QUESTION NO: 48

An app builder wants to update a field on the parent record When a child record connected via lookup is delete Which automation should the app builder use?

A. Record-triggered flow

B. Validation rule

C. Autolaunched flow

D. Quick action

ANSWER: A

Explanation:

Record-triggered flow is the appropriate automation because it can run automatically in response to a record deletion and perform actions on related records. The flow should be configured for the child object, with the trigger set to run when a record is deleted. Because the child record’s lookup value identifies the parent before the deletion completes, the flow can use that relationship to locate the parent record and update the required field, such as a count, status, rollup-related value, or summary indicator. This is a declarative solution that does not require user interaction or custom Apex, making it well suited to an app builder’s requirement. Salesforce’s record-triggered flow configuration supports flows that run when records are created, updated, or deleted. For deletion scenarios, Salesforce recommends using the flow’s available record and prior-value context to reference information from the record being removed and then applying the necessary update to related data. See Salesforce’s Record-Triggered Flows documentation and the Update Records flow element reference.

QUESTION NO: 49

The Director of Customer Service wants to receive a notification when a case stays in the new status for more than four business hours. Which two automation processes should be used to accomplish this? Choose 2 answers

A. Validation Rule

B. Flow Builder

C. Scheduled Apex

D. Escalation rules

ANSWER: B D

Explanation:

Flow Builder can automate follow-up processing for a Case after it enters the New status. A record-triggered flow can be configured with scheduled work that evaluates the Case after the required interval, confirms that its status remains New, and sends a notification to the Director of Customer Service. The flow should be designed so the timing and subsequent status check support the organization’s defined business-hours policy and prevent a notification when the Case has already moved out of New.

Escalation rules are a standard Service Cloud automation feature designed specifically for time-based Case management. An escalation-rule entry can use the organization’s business hours to determine when the four-hour threshold is reached. Its escalation actions can send an email notification to the designated recipient while the Case still meets the escalation criteria. This provides a declarative way to ensure that nonworking time is excluded according to the applicable business-hours

configuration. Salesforce documents escalation rules and their time-based actions at Salesforce Help: Escalation Rules. For Flow scheduling concepts, see Salesforce Help: Scheduled Paths in Record-Triggered Flows.

QUESTION NO: 50

Universal Container's app builder needs to display an account's rating on all contacts related to that account.

Which formula is valid in a text formula field on the contact to display the appropriate value? (Choose two.)

A. CASE(Account.Rating, Hot, Hot, Warm, Warm, Cold, Cold, Not Rated)

B. CASE(Account.Rating, "Hot", "Hot", "Warm", "Warm", "Cold", "Cold", "Not Rated")

C. Account.Rating

D. Text(Account.Rating)

ANSWER: B D

Explanation:

A contact formula can reference a field on its parent Account through the relationship name, so `Account.Rating` accesses the rating associated with each contact's account. Because Rating is a picklist field and the formula's return type is Text, `Text(Account.Rating)` is valid: the `TEXT()` function converts the selected picklist value into text for display. This lets the formula show the account's current rating dynamically on every related contact.

The `CASE(Account.Rating, "Hot", "Hot", "Warm", "Warm", "Cold", "Cold", "Not Rated")` formula is also valid in a text formula field. `CASE()` evaluates the Account Rating value and returns the matching text result; if no listed rating is selected, it returns `"Not Rated"`. This approach is useful when the displayed text may need to differ from the stored picklist value or when an explicit default is required. Both formulas are cross-object formulas, so they recalculate from the related Account data rather than copying rating data onto the Contact. See Salesforce's formula function reference and cross-object formula guidance.

QUESTION NO: 51

An App Builder has been asked to integrate Salesforce with an external web service. The web service must be notified every time an Opportunity is Won. Which two can satisfy this requirement?

A. Use a workflow rule and an outbound message

B. Use a flow and an outbound message

C. Use a process and Apex Code

D. Use a process and an outbound message

ANSWER: A C

Explanation:

Use a workflow rule and an outbound message is a declarative way to notify an external web service when an Opportunity reaches the Won stage. The workflow rule can evaluate criteria such as an Opportunity's Stage being Closed Won, and its outbound-message action sends a SOAP-formatted notification to a configured endpoint. Outbound messaging is designed for this event-notification pattern and includes Salesforce retry behavior when the receiving endpoint is temporarily unavailable. Salesforce documents outbound messaging as a workflow action that sends SOAP messages to external services: Salesforce Outbound Messaging.

Use a process and Apex Code can also meet the requirement. A process can evaluate the Opportunity change and invoke an Apex action. The invoked Apex can make an HTTP callout to the external web service, allowing the integration to use the service's required REST or SOAP interface, authentication mechanism, payload, and endpoint. For process invocation, the

Apex method is exposed with the `@InvocableMethod` annotation; Salesforce describes this mechanism in its Invocable Apex documentation. In production designs, asynchronous Apex is commonly used for callouts so that record processing and external-service availability are appropriately separated.

QUESTION NO: 52

You have a requirement to ensure that if the discount field on an opp is greater than 30%, the record should be automatically submitted for Approval.

Which of the following would help you meet this requirement? (Choose two.)

A. Approval Process

B. Workflow

C. Process Builder

D. Visual Workflow

ANSWER: C D

Explanation:

Process Builder and **Visual Workflow** can evaluate an Opportunity's Discount value and automatically invoke an approval submission when the value exceeds 30%. In Process Builder, the process can be configured to run when an Opportunity is created or edited, use criteria that tests whether the discount is greater than 30%, and execute the *Submit for Approval* action. This submits the record to a configured approval process without requiring a user to click Submit for Approval.

Visual Workflow, now represented by Flow Builder functionality, can implement the same automated pattern. A record-triggered flow can evaluate the Opportunity's discount after the record is saved and use the standard approval-submission capability to submit the record to the applicable approval process. Flow also provides flexibility to add additional decision logic, set submission comments, or dynamically determine an approver when the business process requires it.

For this requirement, the approval process supplies the approval routing and criteria, while automation must perform the actual record submission. Salesforce describes approval-process automation and submission actions in its Approval Processes overview and documents approval actions available to flows in Flow Core Actions.

QUESTION NO: 53

Which three standard component types are available in Lightning App Builder? (Choose three.)

A. Plain text

B. Report details

C. Filter report

D. Rich text

E. Recent items

ANSWER: B D E

Explanation:

Lightning App Builder includes a library of standard components that administrators can drag onto Lightning pages without writing code. **Rich text** is a standard component for placing formatted instructional or informational content directly on a page; it supports content such as headings, links, and images. **Recent items** is also a standard component and provides users with convenient access to records they have recently viewed. **Report details** is available as a standard reporting-related component for displaying report information in a Lightning page context. These components can be selected from the standard component list in the Lightning App Builder and configured as appropriate for the page type and user experience.

Salesforce's Lightning App Builder documentation describes the use of standard components to assemble custom Home, App, Record, and other supported Lightning pages, while the component reference identifies the components provided by Salesforce. See Salesforce Help: Standard Components and Trailhead: Customize Lightning Experience.

QUESTION NO: 54

What objects are supported by the Import Wizard?

A. Contacts

B. Leads

C. Accounts

D. Custom objects

E. Solutions

F. All of the above

ANSWER: F

Explanation:

All of the above is correct because Salesforce's Data Import Wizard supports importing records for accounts, contacts, leads, solutions, and custom objects. The wizard is a browser-based import tool intended for common business-data loads and guided field mapping. It can create new records and, for supported objects, update existing records by matching on an appropriate identifier such as a Salesforce record ID, an external ID, or another supported matching field. When importing accounts and contacts, the wizard also supports the related nature of those records, allowing account and contact data to be loaded in the same import process when the data is prepared appropriately. Custom-object support enables declarative app builders to bring data into objects they create without needing to use the API-based Data Loader for every import. Salesforce documents these supported standard objects and custom objects in its Data Import Wizard guidance. The exact records a user can import or update are also governed by that user's object permissions, field-level access, and sharing-related access. For current object support, import limits, matching behavior, and preparation requirements, see Salesforce's Data Import Wizard documentation and the Trailhead data import module.

QUESTION NO: 55

Northern Trail Outfitters wants the field sales team to only see the accounts that they own Separate North American and European marketing teams should only see accounts m their respective regions. The inside sales team needs to see all accounts in Salesforce.

How can this be accomplished?

A. Set the Organization-Wide Default to Public for accounts. Create criteria-based sharing rules for each marketing team, and create an Inside Sales Team permission set with the "View All" setting for accounts.

B. Set the Organization-Wide Default to Public for accounts. Create profiles for each marketing team, and create an Inside Sales Team role that is at the top of the role hierarchy.

C. Set the Organization-Wide Default to Private for accounts. Create criteria-based sharing rules for each marketing team, and create an Inside Sales Team profile with the "View All" setting for accounts.

D. Set the Organization-Wide Default to Private for accounts. Create permission sets for each marketing team, and create an Inside Sales Team profile with the "View All" setting for accounts.

ANSWER: C

Explanation:

Set the Organization-Wide Default to Private for accounts. Create criteria-based sharing rules for each marketing team, and create an Inside Sales Team profile with the "View All" setting for accounts. This design establishes a restrictive baseline while granting only the additional access required by each business function. With the Account organization-wide default set to Private, account owners retain access to their own records, which meets the field sales team's requirement without broadly exposing accounts to other sales users. Criteria-based sharing rules can then use a regional field on Account, such as Region, to share North American accounts with the North American marketing public group and European accounts with the European marketing public group. This provides targeted record access based on account attributes rather than ownership. Finally, the Account object's View All permission gives inside sales users visibility to every Account record, regardless of ownership or sharing rules. Salesforce documents that organization-wide defaults define the baseline level of record access, after which sharing mechanisms can open access as needed. It also documents that View All permits users to view all records of an object. See Organization-Wide Defaults and User Permissions.

QUESTION NO: 56

Which two objects can be members of a Campaign? (Choose two.)

A. Opportunity

B. Account

C. Lead

D. Contact

ANSWER: C D

Explanation:

Campaign membership in Salesforce is represented by the Campaign Member object. A Campaign Member record associates an individual person with a campaign, and Salesforce supports creating these membership records for both leads and contacts. This lets marketing teams build a unified list of prospective customers and existing contacts who were targeted by, responded to, or otherwise participated in a marketing initiative. Campaign Member records can store a member status, such as Sent, Responded, Registered, or Attended, so teams can track engagement and report on campaign outcomes. Leads can be added directly to campaigns from lead list views, campaign-related actions, imports, or automation. Contacts can be added through the same types of campaign-member tools and are retained as campaign members when a lead is converted, helping preserve campaign-history reporting. Salesforce's campaign documentation describes campaign members as leads and contacts associated with a campaign, and the CampaignMember object reference identifies LeadId and ContactId as the supported person relationships. See Salesforce Help: Add Campaign Members and Salesforce CampaignMember Object Reference.

QUESTION NO: 57

An app builder has a custom component they want to make available on the utility bar, but the component is unavailable. How should the component be tagged?

A. For use in App Manager.

B. For use in Lightning App Builder.

C. For use on record pages.

D. For use on the utility bar.

ANSWER: D

Explanation:

For use on the utility bar. is correct because a custom component must explicitly declare that it supports the utility bar before Salesforce exposes it as an available component in an app's utility bar configuration. For a Lightning Web Component, this declaration is made in the component's `.js-meta.xml` configuration file by including the `lightning__UtilityBar`

target within the <targets> section. That target tells Salesforce the component is intended to run in the utility bar context, allowing an app builder or administrator to add it through Lightning app configuration in App Manager. For example, the metadata includes <target>lightning__UtilityBar</target>. The component can then be selected when editing the app's utility items, subject to the user having appropriate access to the component and app. Salesforce documents the utility-bar target as the target used to make a Lightning Web Component available for utility-bar use. See the Salesforce Lightning Web Components utility bar target documentation and the component configuration file reference.

QUESTION NO: 58

What can a cross-object formula reference?

A. Parent Only

B. Both Parent and Child objects

C. Child Only

D. Records of the same Object

ANSWER: A

Explanation:

Parent Only is correct because a cross-object formula is evaluated from the current record and can traverse an established relationship upward to fields on its parent record. This works for both lookup and master-detail relationships. For example, a formula field on a Contact can reference a field on its related Account, and a formula on a custom detail object can reference a field on its master record. Salesforce supports relationship traversal through multiple parent relationships, allowing formulas to reference fields up to 10 relationships away, subject to formula and relationship limits.

This direction is important because the relationship field stored on the current record identifies its parent, which gives the formula a defined path to that related record's fields. Cross-object references use relationship names and field API names, such as `Account.Industry` from a Contact formula. These formulas can be used in many places where Salesforce supports formulas, helping surface related data without copying it into the current object. Salesforce documents cross-object formulas and their ability to access parent fields in its formula-field guidance: Salesforce Formula Operators and Functions and Salesforce Relationship Fields Reference.

QUESTION NO: 59

An app builder is designing record-triggered flows for a custom object.

Which two statements are true about a before-save record-triggered flow? (Choose two.)

A. It can update fields on the record that triggered the flow.

B. It runs before the record is saved to the database.

C. It can create a related record using a Create Records element.

D. It can send an email alert after the record is saved.

ANSWER: A B

Explanation:

A before-save record-triggered flow can update fields on the record that triggered the flow. This is commonly called a Fast Field Update flow. It updates the triggering record before it is saved and does not require a separate Update Records element for that record.

Before-save flows are designed for efficient updates to the triggering record. They cannot be used to create related records, send email alerts, or update other records. Those actions require an after-save record-triggered flow, which runs after the triggering record has been saved. See Salesforce documentation on Record-Triggered Flows.

QUESTION NO: 60

An app builder has been asked to integrate Salesforce with an external web service. The web service must be notified every time an Opportunity is Won. Which two can satisfy this requirement? (Choose two.)

A. Use a workflow rule and an outbound message.

B. Use Process Builder with an outbound message.

C. Use a flow and an outbound message.

D. Use Process Builder and Apex code.

ANSWER: A D

Explanation:

Using a workflow rule and an outbound message satisfies the requirement because a workflow rule can evaluate an Opportunity when it is created or edited and identify the transition to a Closed Won state. Its outbound message sends a SOAP-based notification containing selected Opportunity fields to a configured external endpoint. Salesforce retains and retries outbound messages when delivery does not initially succeed, making this a supported declarative integration mechanism for notifying an external web service.

Using Process Builder and Apex code also satisfies the requirement. A process can run when an Opportunity meets criteria such as becoming Closed Won, then invoke an Apex action. The Apex implementation can perform the required web-service integration, typically by using an asynchronous callout pattern when appropriate. This approach is suitable when the receiving service requires behavior, authentication, message formatting, or protocol handling beyond the standard outbound-message payload. The process criteria should be configured to run when the record changes to meet the Won condition so that the notification corresponds to the Opportunity becoming won. Salesforce documents outbound messaging as a workflow-based integration feature and supports Apex callouts for programmatic integrations. See Salesforce Outbound Messaging and Apex Callouts.

QUESTION NO: 61

Sales representatives want to capture custom Feedback record details related to each Account. The sales reps want to accomplish this with minimal clicks on the Salesforce Mobile Application. Which two solutions should be recommended in order to meet this requirement? (Choose two.)

A. Create a global action on Account.

B. Create an object-specific action on Account.

C. Create a feedback object as a parent of Account.

D. Create predefined values for most of the fields.

ANSWER: B D

Explanation:

Creating an object-specific action on Account is appropriate because the action is launched from an individual Account record. Salesforce automatically supplies the Account as the context for the action, allowing the new custom Feedback record to be associated with that Account through its relationship field. The action can be placed in the Account page layout's Salesforce Mobile and Lightning Experience Actions section, making it readily available to sales representatives while viewing the relevant Account in the mobile app. This removes the need to navigate separately to Feedback records and manually establish the relationship.

Creating predefined values for most of the fields further reduces the work required on the action form. Predefined values can populate fields on the newly created Feedback record with common defaults, so representatives only need to enter the details that vary for a specific interaction. Used together, the Account-specific action provides the correct record context and predefined field values streamline data entry, directly supporting the requirement for minimal clicks on Salesforce Mobile. Salesforce documents object-specific actions and their record context in its Quick Actions overview, and documents field defaults in Predefined Field Values for Quick Actions.

QUESTION NO: 62

An app builder needs to deploy a new account detail page layout from Sandbox to production. Which three components should an app builder include in the Change Set to ensure it Deploys successfully and visually as expected? Choose 3 answers

A. Detail page layout

B. Custom actions

C. System administrator profile

D. Custom fields

E. Lightning App Builder

ANSWER: A B D

Explanation:

Detail page layout, **Custom actions**, and **Custom fields** should be included in the outbound change set. The detail page layout is the metadata component that defines the Account record-detail experience, including the organization and placement of fields, sections, buttons, quick actions, and related lists. Deploying it makes the new layout definition available in production.

Any custom fields displayed by the layout must also exist in the target organization. Including the relevant custom fields ensures that the deployed layout can render those field components and that users see the intended data-entry and record-viewing experience. Similarly, custom actions used by the layout must be deployed when they are new or do not already exist in production. This preserves the action set presented to users on the Account record page and avoids missing action references after deployment.

Change sets support selecting dependent metadata components, but an app builder should explicitly review the layout's fields and actions and include each required component before uploading and deploying. Salesforce documents change set deployment and component dependencies at Change Sets, and describes layout configuration at Page Layouts.

QUESTION NO: 63

Which statement is true about an External ID field? (Choose two.)

A. The field contains unique record identifiers from a system outside of Salesforce.

B. The field must be unique since duplicates are not allowed within Salesforce.

C. The field must contain at least one number and at least one letter.

D. The field can be unique based on case-sensitive or case-insensitive values.

ANSWER: A D

Explanation:

An External ID field is a custom field designated to store an identifier used by another system, such as a legacy database key, an ERP customer number, or an integration's record ID. This designation enables Salesforce tools and integrations to locate Salesforce records by that outside-system value rather than by the Salesforce record ID. For example, External IDs

are especially useful for upsert operations, where an integration can update an existing record when its outside-system identifier is found or create a record when it is not found.

For text-based External ID fields, Salesforce supports uniqueness settings with either case-sensitive or case-insensitive matching. Case-sensitive matching treats values that differ only in letter case as distinct, while case-insensitive matching treats them as the same value. This lets an app builder align matching behavior with the identifier rules of the source system and maintain dependable integration matching. Salesforce documents External ID fields and their use in record matching and imports in Salesforce Help: External ID Fields and in the Salesforce data import and export Trailhead module.

QUESTION NO: 64

Sales reps at Universal Containers use Salesforce on their mobile devices. They want a way to add new contacts quickly and then follow up later to complete the Additional information necessary. Which mobile solution should an app builder recommend to create the new contact?

A. ? Build a global action to create Contacts.

B. Use Path and set pre-defined values.

C. Customize the mobile menu to move Contacts to the top.

D. Add a compact layout to Contacts.

ANSWER: A

Explanation:

Build a global action to create Contacts is the appropriate mobile solution because a global quick action gives sales reps a fast, focused record-creation experience that is available from the Salesforce mobile app's action interface. An app builder can configure a Create a Record global action for the Contact object and define its action layout to include only the fields needed at the time of initial capture, such as the contact's name, account, phone number, or email address. This keeps the initial interaction short and practical when a rep is working from a mobile device. The rep can save the Contact immediately, then return to the record later to enter the remaining additional information through the normal record-editing experience. Global actions are specifically intended to let users create records from locations that are not tied to a particular object record, making them well suited to quick, on-the-go data entry. Salesforce documents how to create global actions, including Create a Record actions and their layouts, in the Global Quick Actions documentation. For mobile availability and behavior of actions, see Salesforce's Quick Actions in the Salesforce Mobile App overview.

QUESTION NO: 65

Ursa Major Solar (UMS) is planning to hire some new employees. UMS Wants to allow a job candidate (Job Candidate_c) to apply for multiple open Positions (Open_Position c) and then be able to view the applications (Application__c) on the job candidate ' s record UMS also wants to view all the Applications for a specific open position. What should an app builder recommend to meet these requirements? Choose 2 answers

A. Create a master-detail relationship field on Application to Open_Position

B. Create a master-detail relationship field on Application to Job Candidate

C. Create a master-detail relationship on Open_Position to Application

D. Create a master-detail relationship field on Job Candidate to Application

ANSWER: A B

Explanation:

Creating a master-detail relationship field on Application to Open_Position and creating a master-detail relationship field on Application to Job Candidate implement the required many-to-many data model. Application__c serves as the junction object: each application record represents one candidate applying for one particular open position. A candidate can therefore

have multiple related Application__c records, each associated with a different position. Similarly, an open position can have multiple related Application__c records, each associated with a different candidate.

With both master-detail fields defined on Application__c, Salesforce automatically provides related lists on the Job Candidate and Open_Position__c records. Recruiters can view a candidate's submitted applications directly from that candidate's record, and they can view every application received for an opening directly from the open-position record. Master-detail relationships also establish the parent-child ownership and lifecycle behavior appropriate when an application is dependent on its candidate and position records. Salesforce documents that a many-to-many relationship is modeled by creating a junction object with two master-detail relationships to the participating objects. See Salesforce Help: Many-to-Many Relationships and Salesforce Object Reference: Many-to-Many Relationships.

QUESTION NO: 66

A business user wants a quick way to edit a record ' s status and enter a Custom due date field from the record ' s feed in Salesforce Mobile App. What should be used to accomplish this?

A. Custom quick access link

B. Custom action

C. Custom button

D. Custom URL formula field

ANSWER: B

Explanation:

Custom action is the appropriate choice because Salesforce quick actions are designed to let users perform focused tasks directly from supported record contexts, including the Salesforce mobile app. For this requirement, an object-specific update action can be configured for the relevant object and given a layout containing the Status field and the custom Due Date field. When the user opens the action from the record's feed or action bar, Salesforce presents a compact form with only those fields, enabling a fast update without navigating to the full record edit page.

Object-specific actions are contextual: Salesforce automatically associates the action with the record from which it was launched. This makes the action particularly suitable for updating information on that same record, such as its current status and a due date. The administrator can control the action layout, required fields, predefined values, and placement in the object's Salesforce mobile and Lightning Experience actions so that it is readily available to mobile users. This approach provides the streamlined, mobile-friendly record update experience requested. See Salesforce's documentation on object-specific quick actions and quick actions.

QUESTION NO: 67

Duplicate management for Leads has been implemented at Universal Containers but it seems duplicate leads are still being create The Org Wide Default (OWD) is set to " Private " for Leads. Which two actions help prevent duplicate Leads from being created? Choose 2 answers

A. Change the Lead Duplicate Rule details to Bypass Sharing Rules.

B. Change the Lead Assignment Rule to check for duplicates.

C. Change OWD for Leads to Public Read.

D. Change the Lead Duplicate actions to Block on Create.

ANSWER: A D

Explanation:

Changing the Lead Duplicate Rule details to Bypass Sharing Rules ensures that duplicate evaluation can identify matching Lead records even when the user creating the Lead does not have record-level access to those existing records. This is

particularly important when the Lead organization-wide default is Private: visibility restrictions should not prevent the duplicate-management process from comparing the new record with potential duplicates already in the org. The setting enables the duplicate rule to evaluate possible matches across sharing boundaries while preserving normal user access to the underlying Lead records.

Changing the Lead Duplicate actions to Block on Create is also necessary to turn duplicate detection into duplicate prevention. A matching rule identifies likely duplicate records using the selected matching criteria, and the duplicate rule determines the action Salesforce takes when a match is found. Configuring the create action to block prevents the new Lead from being saved when the rule detects a duplicate, rather than merely notifying the user. Together, bypassing sharing rules and blocking on create provide both comprehensive matching and enforced prevention. See Salesforce's Duplicate Management Trailhead module and the Duplicate Rules developer documentation.

QUESTION NO: 68

What is true about Junction objects?

A. Junction object records are deleted when either associated master record is deleted and placed in the Recycle Bin.

B. If both associated master records are deleted, the junction object record is deleted permanently and can't be restored.

C. The first master-detail relationship you create on your junction object becomes the primary relationship.

D. All of the above.

ANSWER: D

Explanation:

All of the above is correct because a junction object is a custom object with two master-detail relationships, used to model a many-to-many relationship. Its records are detail records for both associated master records, so deleting either master record causes the related junction records to be deleted and moved to the Recycle Bin along with the deletion process. This supports the expected cascade-delete behavior of master-detail relationships.

A junction object has special restoration behavior when both of its related master records are deleted. Once both masters are deleted, the junction record is deleted permanently and cannot be restored. Salesforce also designates the first master-detail relationship created on a junction object as its primary relationship. The primary relationship is significant because it determines characteristics such as the junction record's owner and sharing behavior inherited from the primary master. These are standard platform behaviors that an app builder should understand when designing data models involving many-to-many associations. See Salesforce's documentation on many-to-many relationships and junction objects and master-detail relationship considerations.

QUESTION NO: 69

Northern Trail Outfitters wants the field sales team to only see the accounts that they Own. Separate North American and European marketing teams should only see Accounts in their respective regions. The inside sales team needs to see all accounts In Salesforce. How can this be accomplished?

A. Set the Organization-Wide Default to Public for accounts. Create profiles for each marketing team, and create an Inside Sales Team role that is at the top of the Role Hierarchy.

B. Set the Organization Wide Default to Public for accounts. Create criteria-based sharing rules for Each marketing team, and create an Inside Sales Team permission set with the " View All " setting for Accounts.

C. Set the Organization-Wide Default to Private for accounts. Create permission sets for each marketing Team, and create an Inside Sales Team profile with the " View All " setting for accounts.

D. Set the Organization Wide Default to Private for accounts. Create criteria-based sharing rules for Each marketing team, and create an Inside Sales Team profile with the " View All " setting for Accounts.

ANSWER: D

Explanation:

Set the Organization Wide Default to Private for accounts. Create criteria-based sharing rules for Each marketing team, and create an Inside Sales Team profile with the " View All " setting for Accounts. This configuration applies the appropriate baseline and then selectively expands access for each business need. With the Account organization-wide default set to Private, field sales users retain access to Accounts they own but do not receive automatic access to Accounts owned by other users. Criteria-based sharing rules can then share Accounts whose region field meets the relevant North American or European criteria with the corresponding marketing team public group, role, or role-and-subordinates group. This gives each marketing team access based on the Account's regional value without opening every Account to all users. Finally, the View All permission for the Account object gives inside sales users read access to every Account record, irrespective of record ownership, role hierarchy access, or sharing rules. Salesforce documents that organization-wide defaults establish the most restrictive default record access, while sharing rules create automatic exceptions for groups of users. The View All object permission is specifically intended for users who must view all records of an object. See Salesforce organization-wide defaults and Salesforce user permissions.

QUESTION NO: 70

The marketing director is concerned that too many car parts were given Away for free last year. Which functionality should be used to ensure all free parts receive the marketing Directors ' sign-off?

A. Chatter approval

B. Slack post

C. Automated email message

D. Approval process

ANSWER: D

Explanation:

Approval process is the appropriate Salesforce functionality because it provides a formal, enforceable workflow for obtaining the marketing director's authorization before a free-part record can proceed. An approval process can be configured with entry criteria so that it applies specifically when a car part is designated as free. It then submits the relevant record to the marketing director, or to an approver selected through a defined approval step, for approval or rejection.

This declarative feature also supports actions at each stage of the process. For example, the app builder can lock the record while it is awaiting a decision, update a status field after approval, notify the requester, or perform other final approval actions. This creates an auditable approval history showing who reviewed the request and when the decision was made. Consequently, approval process automation directly satisfies the requirement that every free part receive the director's sign-off rather than merely informing the director that a request exists.

Salesforce documentation describes approval processes as automated processes for approving records in Salesforce: Approval Processes Overview. For configuration concepts and approval actions, see Trailhead: Automate Approvals.

QUESTION NO: 71

Which three statements are true about Master-Detail relationships? (Choose three.)

A. Standard objects can be on the detail side of a custom object in a Master-Detail relationship.

B. Master-Detail relationships cannot be converted to a look-up relationship.

C. Deleting a master record in a Master-Detail relationship deletes all related detail records.

D. Master-Detail relationships can convert to a lookup relationship if no roll-up summary fields exist on the master object.

E. A Master-Detail relationship cannot be created if the custom object on the detail side already contains data.

ANSWER: C D E

Explanation:

Deleting a master record in a Master-Detail relationship deletes all related detail records because the detail records are dependent on the master record for ownership and existence. This is the relationship's built-in cascade-delete behavior, which maintains referential integrity for tightly coupled records.

Master-Detail relationships can convert to a lookup relationship if no roll-up summary fields exist on the master object. Salesforce permits this conversion when the master object does not contain roll-up summary fields that rely on the relationship. After conversion, the former detail records become independently owned and the relationship no longer enforces the same dependency behavior.

A Master-Detail relationship cannot be created if the custom object on the detail side already contains data. A new master-detail field is required for every detail record, so Salesforce prevents directly creating this relationship when existing records could lack a master value. Organizations can instead use a lookup relationship and, where applicable, populate it before changing the relationship type. See Salesforce's Object Relationships documentation and Roll-Up Summary Fields documentation.

QUESTION NO: 72

A production org includes custom objects containing confidential Information. A sandbox is needed that includes data records, excludes all of the Confidential objects, and can be refreshed weekly. Which steps should an app builder take to meet these requirements?

A. Create a Developer Pro Sandbox and schedule Data Loader to download selected object data weekly.

B. Create a Full Copy Sandbox and use a sandbox template.

C. Create a Partial Copy Sandbox and use a sandbox template.

D. Create a Developer Sandbox and schedule Data Loader to download selected object data weekly.

ANSWER: C

Explanation:

Create a Partial Copy Sandbox and use a sandbox template. A Partial Copy sandbox is designed for development, testing, and training scenarios that require a representative subset of production data rather than metadata alone. Its sandbox template lets the app builder explicitly select the objects whose records should be copied and, where appropriate, define record-selection criteria. By simply omitting the custom objects that contain confidential information from the template, those objects and their production records are not copied into the sandbox.

This sandbox type also meets the refresh requirement: Partial Copy sandboxes can be refreshed every five days, which supports a weekly refresh cycle. The template is retained for reuse, allowing the same approved data scope to be applied consistently whenever the sandbox is refreshed. Partial Copy sandboxes provide up to 5 GB of data and up to 10,000 records per selected object, so the builder should confirm that the needed nonconfidential test data fits within those limits. Salesforce describes sandbox types and their refresh intervals in its Sandbox Overview documentation and explains data-selection templates in the Salesforce sandbox Trailhead module.