PassQueen
a.k.a dumpsqueen.com

# Salesforce Certification Preparation for Platform App Builder

## Salesforce CRT-402

**Version Demo**

**Total Demo Questions: 34**

**Total Premium Questions: 342**

**Buy Premium PDF**

https://passqueen.com

support@passqueen.com

# Topic Break Down

| Topic | No. of Questions |
|---|---|
| Topic 1, Salesforce Fundamentals | 20 |
| Topic 2, Data Modeling and Management | 99 |
| Topic 3, Security | 49 |
| Topic 4, Business Logic and Process Automation | 80 |
| Topic 5, User Interface | 94 |
| Total | 342 |

QUESTION NO: 1

UniversalContainers has two teams: Sales and Services. Both teams interact with the same records. Sales users use ten fields on the Account record. Services users use three of the same fields as the Sales team, but also have five of their own, which the sales teamdoes not use. What is the minimum configuration necessary to meet this requirement?

**A.** One profile, two record types, one page layout

**B.** Two profiles, two record types, two page layouts

**C.** One profile, one record type, one page layout.

**D.** Two profiles, onerecord type, two page layouts

**ANSWER: D**

**Explanation:**

**Two profiles, onerecord type, two page layouts** is the minimum configuration because both groups work with the same Account records, so separate record types are not needed. A single Account record type can be available to both teams. Two profiles provide separate field-level security settings: the shared Account fields can be available to both teams, while the five Services-specific fields can be made available only to Services users. This ensures that the Sales team cannot access fields that are not intended for them, rather than merely hiding those fields visually.

Two page layouts can then be assigned for the same Account record type, with one layout assigned to the Sales profile and the other to the Services profile. The Sales layout presents the ten fields relevant to Sales, while the Services layout presents its shared fields and Services-specific fields in an appropriate arrangement. Salesforce page-layout assignment supports selecting a page layout by both profile and record type, allowing distinct user experiences without creating record types solely for layout differences. See Salesforce's Page Layouts overview and field-level security documentation.

**QUESTION NO: 2**

Universal Containers has two types of applicants, hourly and salary. There are separate record types for each. Whileall members of the human resource department need to be able to view all applicant records, only the hiring Manager and VP of HR should be able to create salary applicant records.

What should the app builder recommend to meet this requirement?

**A.** Update the org-wide default to private and create a sharing rule for the role of recruiting manager.

**B.** Create a permission set containing the salary record type and assign it to the appropriate users.

**C.** Remove " create " permission for the salary applicant objectfor everyone except the manager and VP.

**D.** Configure the hourly record type as the default and instruct non-management users to accept the default record type.

**ANSWER: B**

**Explanation:**

Create a permission set containing the salary record type and assign it to the appropriate users. Record types control the business processes, picklist values, and page layouts available when users create or edit records. Crucially, record type availability can be assigned through profiles and permission sets. A permission set is the appropriate tool when a limited subset of users needs additional access without changing the baseline access granted to the broader Human Resources group.

In this case, Human Resources users can retain the object permissions needed to view applicant records and create the standard hourly applicant records. The Hiring Manager and VP of HR can receive a permission set that makes the Salary Applicant record type available to them. When these users create an applicant, they can select the salary record type; users who do not have that record type assigned cannot create records using it. This supports least-privilege access and avoids creating unnecessary profile variations solely for two users.

Record type assignment through permission sets is documented in Salesforce's Record Types guidance. For background on using permission sets to grant additional user capabilities, see the Salesforce User Access and Permissions Trailhead module.

## QUESTION NO: 3

An app builder installed a custom Lightning component from AppExchange and has deployed My Domain.

What should be done next in order to configure the component for use in a record page?

**A.** Edit a record page using Lightning App Builder > Drag the component onto the page.

**B.** Edit a record page using the Page Layout editor > Drag the component onto the page.

**C.** Edit a record page using the Page Layout editor > Drag the Visualforce component onto the page.

**D.** Edit a record page using App Manager > Drag the component onto the page.

## ANSWER: A

**Explanation:**

**Edit a record page using Lightning App Builder > Drag the component onto the page.** is correct because Lightning App Builder is the Salesforce tool used to create and customize Lightning record pages. After the AppExchange package is installed and My Domain is deployed, a Lightning component that is exposed for record pages is available in the Lightning App Builder component palette. The app builder opens the applicable object's Lightning record page, drags the custom component onto the desired region, and then configures any component properties that the package developer made available. The page can then be saved and activated for the intended org default, app, record type, or user profile assignments. This is the standard declarative process for making a packaged Lightning component available in the Lightning Experience record-page interface. My Domain is a prerequisite for using Lightning components in Lightning Experience and is therefore appropriately completed before page configuration. Salesforce's guidance on building record pages and adding components is available in the Lightning record page documentation. For the broader process of configuring pages with Lightning App Builder, see Salesforce Trailhead: Customize Record Pages.

## QUESTION NO: 4

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 controllingfield

**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 controllingfields but not dependent fields

## ANSWER: B C E

**Explanation:**

Field dependencies let an administrator restrict the values presented in one field according to the selected value in another field. A custom picklist field can be configured in either role: it can supply the controlling value set or present filtered dependent values. This flexibility makes custom picklists appropriate for modeling related business selections, such as a category followed by a more specific subcategory.

A multiselect picklist is supported as a dependent field. Its available values can therefore be filtered using the selection made in a controlling field, while the user may still select more than one of the permitted dependent values. A checkbox can serve

as a controlling field, using its selected and unselected states to determine which dependent picklist values are available. This is useful when a simple yes-or-no business condition drives the choices a user should see.

These supported field-role combinations allow Universal Containers to guide users toward valid, context-specific values at data-entry time. Salesforce documents the supported controlling and dependent field types in Define Field Dependencies and provides configuration guidance in the Trailhead picklist administration module.

## QUESTION NO: 5

What may occur if workflow rules are re-evaluated after a field change by a field update? Choose 2 answers

**A.** Cross-object workflow rules result in re-evaluation after field change.

**B.** Workflowrules trigger validation rules on field updates

**C.** Workflow rules trigger more workflow rules to be re-evaluated.

**D.** A recursive loop potentially results in exceeding organizational limits.

## ANSWER: C D

**Explanation:**

Workflow rules trigger more workflow rules to be re-evaluated because a workflow field update can cause Salesforce to evaluate applicable workflow rules again when the rule's evaluation criteria are configured to re-evaluate after a field change. The resulting field updates can change values that satisfy the criteria of other workflow rules, allowing further workflow actions to be queued and processed in the same overall transaction. This behavior is important when automation on an object is intentionally chained through field values.

A recursive loop potentially results in exceeding organizational limits when successive workflow field updates repeatedly cause workflow criteria to become true and initiate additional updates. For example, automation that changes a field value in a way that repeatedly meets another rule's re-evaluation criteria can create a cycle of rule processing. Salesforce applies transaction and automation limits, so workflow designs should ensure that each field update moves the record toward a stable final state rather than repeatedly toggling values. See Salesforce's Order of Execution documentation and Workflow Rules documentation for workflow field-update behavior.

## QUESTION NO: 6

At Universal Containers, the VP of Service has requested a visual indicator flag on each case, based on the case priority. High-priority cases should be flagged red, medium-priorityshould be flagged yellow, and low-priority cases should be flagged green. Which formula would accomplish this requirement? Choose 2 answers

**A.** CASE(Priority, "Low", "/img/samples/flag_green.gif", "Medium", "/img/samples/flag_yellow.gif", "High", "/img/samples/flag_red.gif", "/s.gif")

**B.** IMAGE(IF(ISPICKVAL(Priority, "Low"), "/img/samples/flag_green.gif", IF(ISPICKVAL(Priority, "Medium"), "/img/samples/flag_yellow.gif", IF(ISPICKVAL(Priority, "High"), "/img/samples/flag_red.gif", "/s.gif"))), "Priority Flag")

**C.** IF(ISPICKVAL(Priority, "Low"), "/img/samples/flag_green.gif", IF(ISPICKVAL(Priority, "Medium"), "/img/samples/flag_yellow.gif", IF(ISPICKVAL(Priority, "High"), "/img/samples/flag_red.gif", "/s.gif")))

**D.** IMAGE(CASE(Priority, "Low", "/img/samples/flag_green.gif", "Medium", "/img/samples/flag_yellow.gif", "High", "/img/samples/flag_red.gif", "/s.gif"), "Priority Flag")

## ANSWER: B D

**Explanation:**

Both corrected formulas produce an image URL based on the Case Priority picklist value and then pass that URL to the `IMAGE()` function, which is required to render a visual flag rather than simply return text. The nested `IF()` version evaluates

the priority values in sequence and returns the green, yellow, or red sample flag URL as appropriate. Its final fallback value provides a valid image URL when Priority does not match one of the expected values. The `CASE()` version performs the same mapping more compactly: it compares Priority with each stated picklist value, returns the corresponding flag image path, and supplies a default image path for any unmatched value. In both formulas, `"Priority Flag"` is the alternative text argument for `IMAGE()`, supporting accessibility and supplying descriptive text if the image cannot be displayed. Salesforce formula fields support both conditional functions and image rendering, so either approach meets the requirement for a priority-based visual indicator. See Salesforce's formula function reference and formula function documentation.

## QUESTION NO: 7

DreamMouse Realty has a mentorship program that pairs experienced Realtors wrth new Realtors. Each experienced Realtor can mentor one or several new Realtors, and each new Realtor is required to work witha single experienced Realtor they report to for a probationary period.

What type of relationship would an app builder set up to meet this specification?

**A.** Indirect lookup

**B.** Many-to-many

**C.** Master-detail

**D.** Lookup

## ANSWER: D

**Explanation:**

**Lookup** is the appropriate relationship for this mentorship model. A lookup relationship creates a one-to-many association: one experienced Realtor can be referenced by many new Realtor records, while each new Realtor record stores one reference to its assigned mentor. In practice, the app builder can create a self-lookup relationship on the Realtor object, such as a "Mentor" lookup field that points back to another Realtor record. The related list on an experienced Realtor's record then displays all new Realtors assigned to that mentor.

Because the business requirement says that every new Realtor must have exactly one mentor during probation, the lookup field can be configured as required. This retains the flexibility and independent ownership that is generally appropriate for peer or employee-style reporting and mentorship relationships, while enforcing the required assignment at the field level. Salesforce describes lookup relationships as associations that link two objects and support related lists for the referenced records. See Salesforce Help: Lookup Relationships and Trailhead: Learn About Object Relationships.

## QUESTION NO: 8

Universal Containers wants access to Contact records to be determined by access to the related Account. What should the app builder configure for the Contact organization-wide default?

**A.** Controlled by Parent

**B.** Private

**C.** Public Read Only

**D.** Public Read/Write

## ANSWER: A

**Explanation:**

The app builder should set the Contact organization-wide default to **Controlled by Parent**. With this setting, a user's access to a Contact is based on that user's access to the related Account. Users who can access an Account receive the corresponding access to its related Contacts, while users without access to the Account do not gain independent access to

its Contacts through the organization-wide default. This setting supports account-centric sharing without creating separate sharing rules for Contact records. See Salesforce documentation on organization-wide defaults.

## QUESTION NO: 9

The marketing team at UVC has a list of 400 leads it wants to upload to Salesforce. The team need to avoid creating duplicate records. Which two actions should be taken to meet this requirement? Choose 2 answers

**A.** Utilize a Lead Matching Rule and corresponding Duplicate Rule to block newly created duplicate leads.

**B.** Upload the lead list using the import wizard and select a Matching type to prevent duplicate lead creation.

**C.** Use Data Loader's update function to import lead and match to existing records based on e-mail address.

**D.** Enable Duplicate Matching in the Data Management section in Setup and activate the Lead-to_Lead scenario.

## ANSWER: A B

**Explanation:**

Utilize a Lead Matching Rule and corresponding Duplicate Rule to block newly created duplicate leads. Matching rules define the criteria Salesforce uses to identify potential duplicate lead records, such as exact or fuzzy matches on email, name, company, phone, or custom fields. A duplicate rule then specifies the action Salesforce takes when a match is found. Configuring the rule to block creation prevents an import from inserting records that meet the defined duplicate criteria. This provides consistent duplicate protection not only for this upload, but also for future lead creation through supported Salesforce channels.

Upload the lead list using the import wizard and select a Matching type to prevent duplicate lead creation. The Data Import Wizard supports lead imports and provides matching behavior that identifies existing records during import. Selecting an appropriate matching field, commonly email for leads when it is a reliable unique identifier, allows the import process to recognize existing records rather than create another lead for the same person. Together, import matching and active duplicate-management rules provide both import-specific control and org-wide protection. Salesforce documents duplicate management at Duplicate Management and lead importing with the wizard at Data Import Wizard.

## QUESTION NO: 10

Service agents at Ursa Major Solar want a more condensed case view. Service agents also want to be able to modify the associated contact and account records from the case page layout on the Lightning record page.

Which two components should an app builder use to meet these requirements?

Choose 2 answers

**A.** Path

**B.** Rich text

**C.** Related record

**D.** Tabs

## ANSWER: C D

**Explanation:**

**Related record** is appropriate because the Lightning App Builder Related Record component can place a record related through a lookup relationship directly on the Case record page. For a Case, this enables the builder to surface the associated Contact or Account record and configure the component so that agents can access and edit the relevant related-record details without first navigating away from the Case. The component is designed to present a related record's fields in the context of the current record page, supporting the requirement to modify associated records from the case workspace.

**Tabs** is appropriate for creating a more condensed case view. The Tabs component lets an app builder organize page content into separate tabbed sections, such as Case details, related information, and supporting content. Agents can open the section they need while avoiding an unnecessarily long, crowded record page. This improves information density and usability while retaining access to all configured components on the Lightning record page. Salesforce documents Lightning record-page configuration in the Lightning App Builder record page documentation and describes related-record behavior in the Lightning related records component reference.

## QUESTION NO: 11

Universal Containers wants to streamline its data capture process by linking fields together. They wish to do this sothat 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 theymatch 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: B C E

**Explanation:**

“Custom picklist field can be either controlling or dependent field,” “Multi select picklist can be dependent picklist but not controlling fields,” and “Checkbox fields can be controlling fields but not dependent fields” correctly describe Salesforce field-dependency capabilities. A field dependency filters the values presented in a dependent picklist according to the selection made in its controlling field, which helps users enter consistent, contextually valid data. A custom single-select picklist is flexible because it can serve in either role, depending on the dependency being configured. A multi-select picklist is supported as a dependent field, allowing the available selectable values to be filtered, but it cannot drive another field’s available values. A checkbox can act as the controlling field because its two possible states can determine which values are available in a dependent picklist; a checkbox itself cannot be configured as the dependent field. These supported field types let Universal Containers create the requested guided data-capture experience directly with standard Salesforce field dependencies. See Salesforce’s field dependency documentation and the Trailhead unit on controlling and dependent picklists.

## QUESTION NO: 12

Universal Containers uses a custom picklist called Account_Region__c on the Account object. The vice president of

sales has asked that the value of this field is visible on Opportunities.

How should an app builder create this solution?

**A.** Lookup field

**B.** Field-level security

**C.** Field history tacking

**D.** Cross-object formula field

## ANSWER: D

**Explanation:**

Cross-object formula field is the appropriate solution because an Opportunity has a standard relationship to its parent Account. A formula field created on the Opportunity object can traverse that relationship and display the Account’s

`Account_Region__c` value without duplicating or synchronizing data. For a custom picklist source field, the formula field should generally use a Text return type and convert the selected picklist value to text, for example: `TEXT(Account.Account_Region__c)`. The resulting field can then be placed on Opportunity page layouts, Lightning record pages, reports, and other supported interfaces so sales users see the account region while working with an opportunity.

Because formula fields are calculated dynamically, the displayed region always reflects the current value on the related Account. This avoids maintaining automation to copy the value whenever an Account or Opportunity changes, and it preserves a single source of truth for the region. Salesforce describes cross-object formulas as formulas that reference fields on related objects, including parent-object fields available through a relationship. Review Salesforce's cross-object formula field documentation and the Cross-Object Formulas Trailhead unit for implementation details.

## QUESTION NO: 13

Cloud Kicks has a shipment date on each shipment that is sent out. Dispatchers need more details on the day and time the shipment was sent out. The app builder needs to change the current field type that is used from Date to Date/Time.

What should the app builder be aware of when it comes to data already in the system?

**A.** The change will be instant

**B.** Historical data will be updated to 12:00 timestamp.

**C.** The field name will change.

**D.** Data loss will be experienced.

## ANSWER: B

**Explanation:**

Historical data will be updated to 12:00 timestamp. When a custom field is changed from the Date type to the Date/Time type, Salesforce retains the existing calendar-date values rather than deleting the records' field data. Because the original Date field contains no time component, Salesforce supplies a default time of 12:00 AM (midnight) for each existing value as part of the conversion. This gives every legacy shipment record a complete date-and-time value while preserving the original shipment date.

The app builder should also recognize that Date/Time values are stored by Salesforce in Coordinated Universal Time and displayed to users according to each user's time-zone settings. Consequently, users in different time zones can see an adjusted clock time or, near a day boundary, a different displayed calendar date. The newly captured values can include the actual dispatch time, while the converted historical values represent midnight defaults rather than verified historical dispatch times. Salesforce documents the behavior and considerations for custom-field type changes in its Custom Field Types guidance. The platform's treatment of date and date/time data types is also described in the Salesforce Field Types Reference.

## QUESTION NO: 14

Universal Containers (UC) has large data volumes and is nearing data storage limits. The planned solution is to archive historical data to reduce data storage inSalesforce; however, UC would still like to use reports, queries, and lookups on the archived information.

Which two options could meet this requirement?

Choose 2 answers

**A.** Big objects

**B.** Custom objects

**C.** Related objects

**D.** External objects

**ANSWER: A D**

**Explanation:**

**Big objects** are designed for retaining and accessing very large volumes of historical or archival records without consuming the organization's standard data-storage allocation in the same way as ordinary object records. They provide a Salesforce-native archival model and can be queried with SOQL or Async SOQL. Custom big objects can also define lookup relationship fields, allowing archived records to retain relevant associations to Salesforce records. Their index must be planned carefully because it controls how records can be queried efficiently at scale.

**External objects** provide a second valid approach when the historical data is stored outside Salesforce. Salesforce Connect exposes the external data in Salesforce in real time rather than copying it into standard Salesforce data storage. External objects can be accessed with SOQL, can participate in supported external and indirect lookup relationships, and can be used in reporting where supported by the external-data connector and configuration. This lets users work with archived information through Salesforce while the source system remains responsible for storing the large historical dataset.

Salesforce's guidance on large data volumes supports selecting an archival pattern based on whether the organization needs Salesforce-managed archival storage or live access to data maintained externally. See Salesforce Big Objects documentation and Salesforce External Objects documentation.

## QUESTION NO: 15

What the true statement is in regards to converting a tabular, summary, or matrix report to a joined report? Choose 3 answers

**A.** Joined report blocks are formatted as matrix reports.

**B.** Bucket fields are not supported in joined reports.

**C.** Cross filters are not supported in joined reports

**D.** The rows to display filter is not supported in Joined reports

**E.** Report formula fields are nor supported in Joined reports

**ANSWER: B C D**

**Explanation:**

When a report is converted to a joined report, Salesforce applies the feature limitations of the joined-report format. **Bucket fields are not supported in joined reports.** Bucketing is a report-level categorization feature, whereas joined reports are organized into independent report blocks that can use different report types, filters, and groupings. **Cross filters are not supported in joined reports.** Cross filters are therefore not available for defining related-record inclusion or exclusion criteria after the conversion. **The rows to display filter is not supported in Joined reports.** This filter is associated with limiting displayed detail rows and is unavailable in the joined-report format. These constraints are important during report design: administrators should evaluate required bucketing, relationship-based filtering, and row-display behavior before choosing a joined report. Salesforce documents joined reports as a specialized format for comparing data from multiple report types or perspectives, with capabilities and limitations that differ from standard tabular, summary, and matrix reports. See Salesforce's Joined Reports documentation and the Salesforce reports Trailhead module for report-format guidance.

## QUESTION NO: 16

What is a feature that can extend record access beyond the organization-wide defaults? Choose 2 Answers

**A.** Public or private groups.

**B.** Criteria-based sharing rules.

**C.** Owner-based sharing rules.

**D.** Dynamic role hierarchy

**ANSWER: B C**

**Explanation:**

Criteria-based sharing rules and owner-based sharing rules extend record access beyond an object's organization-wide defaults by automatically granting additional access to selected users, roles, territories, or public groups. These rules are configured per object and provide a scalable, declarative way to open access without changing the organization-wide baseline for every record.

Criteria-based sharing rules grant access when a record meets defined field-value conditions. For example, an organization can share all accounts whose type is "Customer" with a particular role or public group. This is useful when access should depend on characteristics of the record rather than its owner.

Owner-based sharing rules grant access according to who owns the record. For example, records owned by users in one role, group, or territory can be shared with another role, group, territory, or queue. This supports predictable collaboration patterns based on ownership and organizational structure. Salesforce evaluates these rules to grant the specified level of access, such as Read Only or Read/Write, in addition to access already available through the organization-wide default and other sharing mechanisms. See Salesforce Sharing Rules and Sharing Records in Salesforce.

## QUESTION NO: 17

Which two solutions prevent a formula field from being referenced by a Roll-Up Summary Field?

**A.** A cross-object workflow updating a field referenced by the formula field

**B.** A cross-object field reference in the formula field

**C.** The CASE() function in the formula field

**D.** The NOW() function in the formula field

**ANSWER: A B**

**Explanation:**

A formula field that includes a cross-object field reference cannot be used as the source field for a roll-up summary. Cross-object formulas obtain a value through an object relationship rather than solely from the detail record itself. Because roll-up summaries aggregate detail-record values and must reliably recalculate as related data changes, Salesforce restricts formula fields that depend on values from another object.

A formula field also cannot be used in a roll-up summary when it references a field updated through a cross-object workflow field update. Such updates can affect a related record outside the immediate detail-record context, creating a dependency that Salesforce does not support for roll-up-summary aggregation. When designing this requirement, use a stored field on the detail object that is populated by supported automation, or redesign the aggregation so that its summarized value depends only on eligible detail-record fields.

These restrictions are part of Salesforce's documented roll-up-summary limitations for formula fields and help ensure summary values remain consistent as records and relationships are updated. See Salesforce's Roll-Up Summary Fields documentation and the Custom Formula Fields documentation.

## QUESTION NO: 18

The CFO of Cloud Kicks needs to sign off on any major show retail deal that has a discount of more than 30% before the deal can be closed.

What feature would be used to handle this requirement?

**A.** Approval Process

**B.** Email Alert

**C.** Field Update

**D.** Workflow Rule

**ANSWER: A**

**Explanation:**

**Approval Process** is the appropriate feature because it routes records that meet specified entry criteria to designated approvers before a business action can be completed. In this case, the process can be configured on the deal record with criteria that identify major show retail deals whose discount exceeds 30%. The CFO can be assigned as the approver, ensuring that qualifying deals are formally reviewed and approved or rejected.

An approval process also supports the required control over closing the deal. Administrators can define initial submission actions, approval steps, final approval actions, and final rejection actions. For example, the record can be locked while approval is pending, and final approval can update a status or other field that permits the deal to proceed to closure. This provides an auditable approval history, including the submission, approver decisions, comments, and timestamps, which is important for financial oversight and discount governance.

Salesforce approval processes are designed specifically for structured, multistep business approvals based on record data and can include automated actions at each stage. See Salesforce's Approval Processes overview and Approval Processes Trailhead module for configuration concepts and examples.

**QUESTION NO: 19**

DreamHouse Realty wants to make sure an Opportunity has a fieldExpected_Close_Date_c populated before it is allowed to enter the qualified stage.

How should an app builder solution this request?

**A.** Record Type

**B.** Validation Rule

**C.** Activity History

**D.** Page Layout

**ANSWER: B**

**Explanation:**

**Validation Rule** is the appropriate solution because it enforces data-quality requirements whenever an Opportunity is saved, regardless of whether the record is updated through the standard user interface, a mobile client, an import, an integration, or automation. The app builder can create a rule that evaluates whether the Opportunity Stage equals Qualified and whether `fieldExpected_Close_Date_c` is blank. When both conditions are true, the validation rule prevents the save and presents a clear error message directing the user to populate the expected close-date field. A formula such as `AND(ISPICKVAL(StageName, "Qualified"), ISBLANK(fieldExpected_Close_Date_c))` expresses this requirement. This approach permits users to work with Opportunities in earlier stages without requiring the field, but makes the field mandatory at the precise point the business process requires it. Salesforce validation rules are designed to verify that entered data meets specified standards before records can be saved, making them the declarative mechanism for conditional field requirements. See Salesforce Validation Rules documentation and Trailhead: Validation Rules.

**QUESTION NO: 20**

The VP of Marketing wants to broadcast an email to 10,000 contacts in Salesforce on a regular basis, but realizes Salesforce's mass email functionality has a limitation on the number of emails that can be sent each day.

**A.** Develop Apex code and Visualforce pages to send the emails.

**B.** Download all contacts to a CSV file and use an email client to send the mails

**C.** Research and evaluate products available on AppExchange to send mass emails

**D.** Request Salesforce increase the number of maximum daily mails.

**ANSWER: C**

**Explanation:**

**Research and evaluate products available on AppExchange to send mass emails** is correct because a recurring marketing campaign to 10,000 contacts requires a purpose-built bulk-email or marketing-automation solution rather than standard Salesforce mass email. Salesforce applies daily email limits to protect platform reliability and email deliverability, so an organization should select a product designed to manage high-volume commercial email sends. AppExchange offerings can integrate with Salesforce contact and campaign data while providing capabilities important for ongoing marketing operations, such as audience segmentation, subscription and consent management, reusable templates, scheduled sends, bounce handling, delivery reporting, and unsubscribe processing. This approach also helps the marketing team work within applicable anti-spam and privacy requirements by using tools intended for permission-based marketing communications. The AppExchange marketplace provides products that extend Salesforce and can be assessed for the organization's volume, integration, compliance, and reporting needs. Salesforce documents that email sending is subject to limits, which should be considered when designing a solution for large-scale communications. See Salesforce's Mass Email Limits documentation and the Salesforce AppExchange for available integrated solutions.

**QUESTION NO: 21**

What can be configured in a report subscription? Choose 2 answers

**A.** Schedule when the report is run and sent to recipients.

**B.** Set conditions based on report results for sending notifications.

**C.** Change the report type used by the report.

**D.** Grant recipients access to records returned by the report.

**ANSWER: A B**

**Explanation:**

A report subscription can be configured with a schedule that determines when Salesforce runs the report and sends the results to the selected recipients. For example, a manager can receive a pipeline report every Monday morning without manually running it.

Report subscriptions can also include conditions based on report results. When the selected conditions are met, Salesforce sends the report notification according to the subscription configuration. This supports exception-based reporting, such as notifying a sales manager only when the total pipeline drops below a specified value. See Salesforce documentation on subscribing to reports.

**QUESTION NO: 22**

An app builder at DreamHouse Realty created a custom object which has fields containing data from two different objects via related lookups.

What is needed to create "with" or "without* reports on the new custom object?

**A.** Row-Level Formula

**B.** Report Bucket Field

**C.** Report Filters

**D.** Custom Report Type

**ANSWER: D**

**Explanation:**

Custom Report Type is correct because it defines the object relationships and record inclusion behavior available when users create reports. A custom report type starts with a primary object, such as the new custom object, and can include related objects through the configured relationship structure. When defining that structure, the app builder can choose whether the report returns only primary-object records that have related records ("with") or includes primary-object records regardless of whether a related record exists ("with or without"). This configuration is what makes the appropriate reporting scope available to report authors; it is established before an individual report is built. Custom report types also control which fields from the selected objects are exposed for reporting, allowing fields from the custom object and its related lookup objects to be used together in the report builder. After deployment, users select the custom report type and add filters, groupings, and columns as needed for the business question. Salesforce describes custom report types as templates that determine the objects, relationships, and fields available for reports. See Salesforce's Custom Report Types documentation and the Trailhead unit on custom report types.

## QUESTION NO: 23

Which three Salesforce functionalities are ignored when processing field updates in workflow rules andapproval processes?

**A.** Multiple currencies

**B.** Field-Level Security

**C.** Validation Rules

**D.** Record type picklist value assignments

**E.** Decimal places and character limits

**ANSWER: A D E**

**Explanation:**

**Multiple currencies**, **Record type picklist value assignments**, and **Decimal places and character limits** are ignored during field updates performed by workflow rules and approval processes. These updates run as system automation rather than as an interactive user entering values through the standard record-editing experience. Consequently, Salesforce applies the field-update action without using the normal multiple-currency behavior as a constraint on the update. A field update can also assign a picklist value even when that value is not assigned to the record's current record type. This is useful when automation must enforce a business-process value independently of the picklist choices presented to users for a particular record type.

Likewise, workflow and approval field updates can process assigned values without enforcing ordinary decimal-place and character-limit constraints in the way interactive edits do. Administrators should therefore design these actions carefully, particularly when an automated value is consumed by downstream formulas, integrations, reports, or user interfaces. These behaviors are important when maintaining legacy workflow automation and approval-process actions, even though Salesforce generally recommends Flow for new automation. Salesforce documents workflow field-update behavior and its automation context in its workflow and approval documentation: Salesforce Help: Workflow Field Updates and Salesforce Help: Approval Processes Overview.

## QUESTION NO: 24

An app builder noticesseveral Accounts converted from Leads are missing information they expected to be caught via Account validation rules.

What could be the source of this issue?

**A.** The lead settings are unchecked to require validation for converted leads.

**B.** Account validation rules fail to validate on records converted from a lead.

**C.** The lead settings are allowing users to intentionally bypass validation rules.

**D.** Lead validation rules fail to validate on records when they are being converted.

**ANSWER: B**

**Explanation:**

**Account validation rules fail to validate on records converted from a lead.** Lead conversion is a specialized Salesforce process that creates or updates related Account, Contact, and optionally Opportunity records from the Lead. An app builder should not assume that Account validation rules will provide the same enforcement during this conversion pathway as they do when users create or edit an Account directly. Consequently, a converted Account can be created with missing information that would ordinarily be identified by an Account validation rule in a standard Account save operation. This behavior is important when the organization relies on Account-level data-quality requirements for values mapped from Lead fields. The appropriate design is to account for lead-conversion behavior explicitly and enforce critical conversion requirements using the supported controls for the Lead conversion process, suitable automation, or a conversion-aware implementation strategy. Salesforce documents lead conversion as a distinct operation with its own considerations and settings, including validation behavior. See Salesforce Help's Convert Leads guidance and the Salesforce Trailhead module on converting leads in Lightning Experience.

## QUESTION NO: 25

The customer service team at Universal containers uses a custom Project object to track their customer projects. An App builder needs to create a relationship between the project object and the related client Account record. A private sharing model is in place for both Accounts and projects. Which statement is true when creating an Account lookup field on the Project object? Choose 2 answers

**A.** The account lookup field on the project record can be made optional.

**B.** Cross-object field updates between the Project and Account records are not supported

**C.** Users can only delete Accounts if they have access to related projects records

**D.** The account record can include roll-up summary field with data from the project records

**ANSWER: A C**

**Explanation:**

The account lookup field on the project record can be made optional. A lookup relationship does not inherently require every Project record to reference an Account; the field can be configured so that users may save a Project without an Account value. This supports situations such as projects that are still being qualified or have not yet been assigned to a client. Salesforce describes lookup relationships as flexible links between records, unlike relationships where the child record must always have a parent.

Users can only delete Accounts if they have access to related projects records. Because deleting an Account can affect the relationship values held by related Project records, access to those related records is relevant to whether Salesforce permits the delete operation. This is especially important when Account and Project sharing are private: ownership or access to an Account alone does not automatically grant access to the related Project records. The administrator should consider sharing rules and user access whenever users need to manage Account records that have related Projects. See Salesforce's relationship data-modeling guidance and record-sharing guidance.

## QUESTION NO: 26

Ursa Major Solar's service department gets requests for several types of services, such as installation, repair, and maintenance. Service managers need to be able totell when maintenance was last done on on asset to help determine If they are meeting contract agreements, but the last maintenance date can be difficult to determine when there are many work orders related to the asset. They think it would be helpful to have a field auto-populated on the Asset record when a maintenance work order gets closed.

What tool should an app builder recommend to help meet this requirement?

**A.** Visualforce

**B.** Roll-up Summary

**C.** Apex Trigger

**D.** Flow

**ANSWER: D**

**Explanation:**

**Flow** is the appropriate declarative automation tool for maintaining a date on the related Asset when a maintenance Work Order closes. An app builder can create a record-triggered flow on the Work Order object that runs when a record is updated, evaluates whether the Work Order is a maintenance request and has moved to a closed status, and then updates the related Asset's custom Last Maintenance Date field. The flow can set that field from the Work Order's completion or close date, providing service managers with a single, immediately visible value on the Asset rather than requiring them to review a long history of related Work Orders.

This approach also supports the requirement without custom code and can be configured to run only when the relevant status changes, which helps avoid unnecessary updates. A record-triggered flow is specifically designed to automate actions in response to record creation or updates, including updating fields on related records. Salesforce recommends using Flow Builder for this type of no-code business-process automation. See Salesforce's record-triggered flow documentation and Record-Triggered Flows Trailhead module.

**QUESTION NO: 27**

Universal container has a custom object for shipping information. They have to ship to both businesses and consumers. They need to show additional values in the custom field called insurance type for business shipping records. How can this be set up?

**A.** Use Record type with single page layout.

**B.** Create multiple picklist fields on the object

**C.** Use record types with multiple page layouts

**D.** Create a multi-select pick-list field.

**ANSWER: C**

**Explanation:**

Use record types with multiple page layouts. Record types are designed to support distinct business processes on the same object, such as business shipments and consumer shipments. For the shipping custom object, an administrator can create separate record types for these two kinds of records and configure the Insurance Type picklist's available values for each record type. The business-shipping record type can include the additional insurance values, while the consumer-shipping record type can expose only the values relevant to consumers. This keeps the data model simple because there is still one Insurance Type field, while users see a context-appropriate set of choices when creating or editing a record.

Record types can also be assigned to page layouts. This allows the business-shipping experience to display fields and sections needed for commercial shipping, while the consumer-shipping layout can be tailored to consumer shipments. Salesforce supports assigning picklist values by record type and associating record types with page layouts, making this the appropriate declarative configuration for different shipping audiences. See Salesforce's Record Types documentation and Picklist Values and Record Types documentation.

**QUESTION NO: 28**

What can be configured using Dynamic Forms on a Lightning record page? Choose 2 answers

**A.** Place individual fields and field sections on a Lightning record page.

**B.** Configure visibility rules for fields and field sections.

**C.** Change the organization-wide default sharing setting for the object.

**D.** Create a new record type for the object.

**ANSWER: A B**

**Explanation:**

Dynamic Forms allows an app builder to place individual fields and field sections directly onto a Lightning record page. This provides more flexibility than relying only on the fields presented through the Record Detail component and assigned page layout.

Dynamic Forms also supports component visibility rules for fields and field sections. For example, an app builder can display a renewal section only when an Account has a Customer Type of Existing Customer. Visibility rules help present relevant information while reducing unnecessary fields for users. See Salesforce documentation on Dynamic Forms.

## QUESTION NO: 29

Universal Containers has a customer base where many customers have the same or similar company names.

Which functionality should be configured to improve an end user`s search experience?

Choose 2 answers

**A.** Update the account search layouts view filtersettings.

**B.** Update the account search layouts search results columnsdisplayed.

**C.** Update the account search filter fields.

**D.** Update the account search layouts accounts tab columns displayed.

**ANSWER: B C**

**Explanation:**

Update the account search layouts search results columnsdisplayed and update the account search filter fields. When many Account records have alike names, users need meaningful context in the search-result list to identify the intended company without opening multiple records. Configuring the Account Search Results layout lets an administrator choose the fields shown beside each matching account, such as account owner, phone, billing city, type, industry, or parent account. These details make otherwise similar company names distinguishable at a glance.

Configuring Account Search Filter Fields complements the result columns by making selected Account fields available as filters on the search-results page. Users can then narrow a broad name search using relevant business attributes, for example account type, industry, location, or ownership. Together, informative result columns and targeted filters reduce ambiguity, shorten the time needed to locate the correct record, and make global search more practical for a large customer base with duplicate or near-duplicate names. Salesforce documents that search layouts control the fields displayed for search results and identifies search filter fields as the fields users can use to refine search results. See Customize Search Layouts and Search Filter Fields.

## QUESTION NO: 30

A custom field contains a feedback score which is on a scale of one to five. End users would like a visual indicator of one to five stars based on the number found in the feedback score custom field. How can this visual indicator be displayed?

**A.** Use a custom formula field.

**B.** Use a custom image field.

**C.** Use a custom number field.

**D.** Use a custom text field

**ANSWER: A**

**Explanation:**

Use a custom formula field. A formula field can evaluate the numeric feedback score and return a visual result rather than merely displaying the stored number. For this requirement, the formula can use Salesforce's `IMAGE()` function to render star-image resources, with conditional logic such as `CASE()` or nested `IF()` statements selecting the appropriate one-star through five-star image based on the score. The formula's return type should be Text so that the image markup can be rendered in supported Salesforce user interfaces. This approach is declarative, requires no automation to keep a second field synchronized, and recalculates automatically whenever the feedback score changes. The image files must be made available to Salesforce, such as through Documents, Static Resources, or another supported image location, and the formula references the relevant image URL and alternate text. Salesforce documents the `IMAGE()` formula function and its use for displaying images in formula fields in the Salesforce Formula Operators and Functions reference. Additional guidance on creating and using formula fields is available in Salesforce Help: Formula Fields.

## QUESTION NO: 31

An app builder at Universal Containers has been asked to add the Chatter feed to a custom object record page.

Which approach should the app builder use?

**A.** Add the standard Chatter feed component.

**B.** Add the standard related list component

**C.** Add a custom Chatter feed component

**D.** Add the Chatter feed component from the AppExchange.

**ANSWER: A**

**Explanation:**

**Add the standard Chatter feed component.** Salesforce provides a standard Chatter component for Lightning record pages, including pages for custom objects. An app builder can open the custom object's record page in Lightning App Builder, drag the Chatter component onto the page, and save and activate the page as needed. This places the record's feed directly on the page so that users can view posts, publish updates, comment, and collaborate in the context of that specific record.

The component uses Salesforce's native Chatter capabilities and honors the organization's Chatter setup, object feed tracking configuration, record access, and user permissions. For a custom object, feed tracking must be enabled for the object before users can meaningfully post to and follow the record feed. The standard component is therefore the supported declarative approach: it requires no custom development and does not require installing a third-party package. Salesforce documents the Lightning Chatter feed component and its use on record pages in the Lightning Component Reference. See also Salesforce Trailhead's overview of Chatter in Lightning Experience.

## QUESTION NO: 32

The Director of Customer Service wants to know when agents are overwhelmed with high-priority items in the support queue. TheDirector wants to receive a notification when a new case is open with the status of "New" for more than four business hours. Which automation process could be used to accomplish this? Choose 2 answers

**A.** Escalation rules

**B.** Visual workflow

**C.** Lightning Process Builder

**D.** Scheduled Apex

**ANSWER: A D**

**Explanation:**

Escalation rules are designed specifically for case-aging requirements. An administrator can define rule criteria that identify the relevant cases, such as high-priority cases whose status remains New, and create an escalation action set to occur after four business hours. Salesforce evaluates the escalation age against the configured business hours, so nonbusiness time can be excluded. The escalation action can send an email notification to the Director, queue members, or another designated recipient. Salesforce documents case escalation configuration at Escalation Rules.

Scheduled Apex can also meet this requirement when custom automation is appropriate. Apex can run on a recurring schedule, locate cases that are still New, and use the BusinessHours methods to determine whether more than four business hours have elapsed rather than merely measuring elapsed calendar time. Once the threshold is met, the scheduled job can send an email or initiate another notification mechanism. The `BusinessHours.diff` and related methods support calculations against the organization's business-hours configuration; Salesforce documents these APIs in the BusinessHours Class reference. This approach is particularly useful if the notification criteria or recipient-selection logic requires custom logic beyond standard escalation configuration.

**QUESTION NO: 33**

Which statement is true for embedding aVisualforce page in a page Layout. Choose 2 Answers

**A.** Visualforce Pages on a field set have attributes for width and height.

**B.** Visualforce Pages can be placed anywhere in the page layout.

**C.** Visualforce Pages on a page layout have attributes for width and height

**D.** Visualforce Pages can only be place in the Visualforce section in a page layout.

**ANSWER: B C**

**Explanation:**

Visualforce Pages can be placed anywhere in the page layout because, after a Visualforce page is made available for the relevant object, an administrator can add it from the Visualforce Pages palette in the page-layout editor and position the resulting section in the layout. This permits the embedded page to appear in the location that best supports the record-viewing workflow, such as alongside standard detail sections or related information.

Visualforce Pages on a page layout have attributes for width and height because the page-layout component exposes display properties for the embedded Visualforce page. These settings control the dimensions of the Visualforce frame as it appears within the record page, allowing the administrator to provide sufficient space for the page's content and avoid unnecessary scrolling or unused area. The Visualforce page must be configured for use in the applicable context, and the page-layout configuration governs its placement and displayed dimensions. Salesforce's Visualforce documentation describes making pages available for use in page layouts, while the page-layout documentation covers configuring Visualforce page components and their display properties. See Salesforce Visualforce Developer Guide and Salesforce Help: Page Layouts.

**QUESTION NO: 34**

What is recommended to refresh a full sandbox?

**A.** After a UAT sign-off

**B.** Whenever a new production user is added.

**C.** After a major productionrelease.

**D.** Within 3 hours of when it is needed.

**ANSWER: C**

**Explanation:**

Refreshing a full sandbox after a major production release is recommended because a full sandbox is intended to closely mirror the production organization, including its metadata and a full copy of production data. A significant production deployment can introduce new fields, objects, automations, integrations, permission changes, and application behavior. Refreshing after that release realigns the sandbox with the current production baseline so that subsequent regression testing, integration testing, performance testing, and release planning are performed against an accurate representation of the live environment.

Because a full sandbox refresh copies production configuration and data, teams should plan the activity carefully, preserve any needed sandbox-only work, and complete any post-refresh tasks such as activating users, updating integration endpoints, and masking or securing sensitive data as required. Full sandboxes have a 29-day refresh interval, so they are best used as part of a deliberate release and testing strategy rather than for frequent, immediate updates. Salesforce describes sandbox types and their refresh intervals in its Sandbox Overview documentation; see also the Salesforce Sandboxes Trailhead module.