



*Up-to-date Questions and Answers from authentic resources to improve knowledge and pass the exam at very first attempt. ---- Guaranteed.*



*Salesforce-Certified-Advanced-Administrator MCQs  
Salesforce-Certified-Advanced-Administrator TestPrep  
Salesforce-Certified-Advanced-Administrator Study Guide  
Salesforce-Certified-Advanced-Administrator Practice Test  
Salesforce-Certified-Advanced-Administrator Exam Questions*



[killexams.com](http://killexams.com)

**Salesforce**

**Salesforce-Certified-Advanced-Administrator**

*Certified Advanced Administrator*

ORDER FULL VERSION

<https://killexams.com/pass4sure/exam-detail/Salesforce-Certified-Advanced-Administrator>



**Question: 450**

A company needs to enforce GDPR-compliant data deletion for expired customer records, with a report tracking deletion requests. What should the administrator implement?

- A. Build a Process Builder to manage deletions and logs
- B. Create an Apex trigger to handle deletions and reporting
- C. Configure a workflow rule to delete records and update reports
- D. Use a Flow to delete records and log requests in a custom object

Answer: D

Explanation: A Flow can identify expired records, delete them per GDPR, and log requests in a custom object for reporting. Apex triggers are less maintainable, workflow rules can't handle logging, and Process Builder is deprecated for new automations.

**Question: 451**

A company uses a custom object to track vendor contracts with a lookup to Accounts. The administrator needs to ensure that contract records are only editable by the Account owner and users in a specific public group. How can this be configured?

- A. Write an Apex sharing rule to grant edit access to the required users
- B. Set OWD to Private and create a sharing rule for the Account owner and public group
- C. Use a permission set to assign edit access to the public group
- D. Configure a manual sharing rule for each contract record

Answer: B

Explanation: Setting the Organization-Wide Default to Private and creating a sharing rule to grant edit access to the Account owner and the public group is the most efficient solution. This leverages standard sharing mechanisms to meet the requirement without custom code or manual processes.

**Question: 452**

An administrator must ensure that a custom object's records are only visible to users in specific territories, with visibility adjusting dynamically based on territory realignment. What should the administrator implement?

- A. Use manual sharing with a Flow to adjust visibility
- B. Configure Enterprise Territory Management with restriction rules
- C. Create an Apex sharing rule with territory logic
- D. Build a permission set with dynamic territory assignments

Answer: B

Explanation: Enterprise Territory Management dynamically adjusts record visibility based on territory assignments, and restriction rules enforce access. Manual sharing is unscalable, Apex sharing is complex, and permission sets don't handle territory-based visibility natively.

**Question: 453**

A retail company uses a custom object, Inventory\_\_c, with a field, Quantity\_\_c. The administrator needs to prevent negative quantities during bulk updates and log attempts to a custom Log\_\_c object. How should this be enforced?

- A. Use a validation rule to check Quantity\_\_c and a Flow to log violations
- B. Configure Process Builder to evaluate Quantity\_\_c and create Log\_\_c records
- C. Write an Apex trigger to validate Quantity\_\_c and insert Log\_\_c records
- D. Use a Flow to validate quantities and trigger a workflow to log attempts

Answer: C

Explanation: An Apex trigger on Inventory\_\_c can validate Quantity\_\_c before save, throwing an error for negative values and inserting a Log\_\_c record for each attempt, including during bulk updates. Validation rules cannot log to custom objects, and Flows/Process Builder are less efficient for bulk operations.

**Question: 454**

DreamHouse Realty needs to migrate millions of legacy records into Salesforce, ensuring data integrity and minimal downtime. Which three steps should the administrator take? Choose 3 answers

- A. Use Bulk API for large-scale imports
- B. Validate data mappings in a sandbox
- C. Implement Flow for data cleansing
- D. Schedule imports during off-peak hours
- E. Enable Change Data Capture for tracking
- F. Use Apex for real-time validation

Answer: A, B, D

Explanation: The Bulk API efficiently handles millions of records, minimizing processing time. Validating mappings in a sandbox ensures accuracy before production import. Scheduling imports during off-peak hours reduces system load, avoiding downtime. Flow is less suited for large-scale cleansing compared to ETL tools. Change Data Capture tracks changes, not migrations. Apex for validation is slower than pre-import checks.

**Question: 455**

An organization uses Salesforce Shield to comply with strict data privacy regulations. The administrator must ensure that only authorized users can view sensitive fields in a custom object after a record is locked, while maintaining Event Monitoring for compliance audits. Which configuration meets these requirements?

- A. Encrypted Custom Fields, Role Hierarchy, and Field Audit Trail
- B. Platform Encryption, Dynamic Forms, and Login Forensics
- C. Field-Level Security, Record Type restrictions, and Transaction Security Policies
- D. Data Masking, Page Layouts, and Event Monitoring settings

Answer: A

Explanation: Encrypted Custom Fields with Platform Encryption (part of Salesforce Shield) protect sensitive data, ensuring only authorized users with the "View Encrypted Data" permission can access it. Role Hierarchy controls record access post-locking, and Field Audit Trail (Shield component) tracks field changes for compliance, complementing Event Monitoring for audit logs. Field-Level Security doesn't encrypt data, and Transaction Security Policies are unrelated to field visibility. Dynamic Forms are UI-based, not security-focused. Data Masking is for sandbox use, not production compliance.

**Question: 456**

An administrator needs to ensure that a custom object, Task\_\_c, is only accessible to users who are assigned to the related Project\_\_c record via a custom junction object. The solution must avoid manual sharing. How should this be configured?

- A. Create a sharing rule based on Project\_\_c assignment
- B. Set OWD to Private and use Apex sharing for dynamic access
- C. Use a Flow to grant access to Task\_\_c records
- D. Assign a permission set to users based on Project\_\_c

Answer: B

Explanation: Setting OWD to Private and using Apex sharing ensures dynamic access to Task\_\_c based on Project\_\_c assignments, avoiding manual sharing. Sharing rules cannot target junction relationships, Flow is inefficient for access control, and permission sets are static.

**Question: 457**

A nonprofit needs to track donor pledges across multiple campaigns, with reports showing pledge fulfillment status aggregated by donor and campaign type. What should the administrator configure?

- A. Create a custom report type with cross-filters for pledges and campaigns
- B. Use a joined report with bucket fields for status aggregation
- C. Build a Visualforce page with Apex for custom reporting
- D. Configure a standard report with formula fields for aggregation

Answer: A

Explanation: A custom report type with cross-filters allows aggregation of pledge data by donor and campaign type, capturing fulfillment status. Joined reports are less flexible, Visualforce is unnecessary, and standard reports can't handle complex cross-object aggregation.

**Question: 458**

At Cloud Kicks, a custom object Project\_\_c has a master-detail relationship with Account. The administrator needs to ensure that when a Project\_\_c record is created, a related Task is automatically generated with a due date 7 days from the Project\_\_c.CreatedDate and assigned to the Account.OwnerId. The solution must handle bulk inserts of over 200 records efficiently. Which approach should the administrator use?

- A. Create a Process Builder to invoke a Flow that creates the Task
- B. Use a Workflow Rule with a Time-Based Action
- C. Write an Apex trigger on Project\_\_c to create the Task
- D. Configure a Scheduled Flow to run daily and check for new Project\_\_c records

Answer: C

Explanation: An Apex trigger is the most efficient solution for this requirement, as it can handle bulk inserts (over 200 records) natively within the transaction, ensuring performance and governor limit compliance. The trigger can use after insert to access CreatedDate and Account.OwnerId, creating Task records in a bulkified manner (e.g., collecting Tasks in a List and inserting once). Process Builder may hit governor limits with bulk operations due to Flow invocation overhead. Workflow Rules don't support dynamic due date calculations based on CreatedDate without complex workarounds. Scheduled Flows introduce delays and unnecessary complexity for an immediate requirement.

### Question: 459

Northern Trail Outfitters uses a complex approval process for opportunities, involving multiple approvers based on dynamic criteria. An administrator notices that some users receive approval requests but cannot submit responses due to an error. What should the administrator investigate first?

- A. The approval process's entry criteria for user eligibility
- B. The users' permission sets for approval process access
- C. The debug log for Apex triggers related to approvals
- D. The process builder's logic for assigning approvers

Answer: B

Explanation: If users receive approval requests but cannot submit responses, the issue likely stems from insufficient permissions, such as lacking the "Submit for Approval" permission or access to the approval process, which is controlled via permission sets or profiles. Investigating permission sets is the most direct step. Entry criteria determine who enters the process, not response submission. Debug logs are useful for code-related issues, but approvals are typically declarative. Process builder logic may assign approvers but doesn't govern response submission.

### Question: 460

Universal Containers needs to enforce that `OrderItem.UnitPrice` cannot be changed after `Order.Status = 'Activated'` unless the user has the `Override_Prices` permission. The solution must handle bulk updates and API calls. Which implementation is most effective?

- A. Apex Trigger with Permission check
- B. Validation Rule with `ISCHANGED` function
- C. Record-Triggered Flow with Decision elements
- D. Process Builder with a Validation action

Answer: A

Explanation: An Apex Trigger on `OrderItem` can check `FeatureManagement.checkPermission('Override_Prices')` and prevent `UnitPrice` changes if `Order.Status = 'Activated'` using `Trigger.newMap` and `Trigger.oldMap`. Triggers handle bulk updates and API calls efficiently. Validation Rules can't check permissions or access parent `Order` data dynamically. Flows lack permission-checking for field-specific logic. Process Builder is deprecated and unsuitable for complex validation.

**Question: 461**

Universal Containers has a global sales team with localized Opportunity processes. The administrator needs to ensure that when an Opportunity is created in a specific region, a custom object is automatically created with region-specific defaults, but only if the Opportunity meets dynamic criteria. What should be used?

- A. Configure a Flow with decision elements
- B. Write an Apex trigger with conditional logic
- C. Use a Process Builder with regional actions
- D. Set up a Workflow Rule with field updates

Answer: A

Explanation: A Flow with decision elements can evaluate dynamic criteria and create records with region-specific defaults without coding. Apex triggers are viable but require maintenance, Process Builder is deprecated, and Workflow Rules lack the flexibility for complex automation.

**Question: 462**

An administrator is configuring a flow to automate opportunity team member assignments based on a custom object Territory\_\_c with fields Region\_\_c and Team\_Member\_\_c (lookup to User). The flow must add team members only if the opportunity's Amount exceeds \$100,000 and Region\_\_c matches the opportunity's Account.Region\_\_c. Which elements are required? Choose 2 answers

- A. A Get Records element to retrieve Territory\_\_c records where Region\_\_c equals Account.Region\_\_c
- B. A Decision element to check if Amount is greater than 100,000
- C. An Assignment element to update OpportunityTeamMember fields directly
- D. A Create Records element to add OpportunityTeamMember records

Answer: A, D

Explanation: A Get Records element fetches Territory\_\_c records matching Account.Region\_\_c to identify the correct Team\_Member\_\_c. A Create Records element adds OpportunityTeamMember records for the team assignments.

**Question: 463**

Northern Trail Outfitters has a custom metadata type `Integration_Settings__mdt` to store API endpoint configurations. An Apex class retrieves these settings using `Integration_Settings__mdt.getInstance('Primary_Endpoint').Endpoint_URL__c`. During a deployment, the administrator notices that the class throws a `System.NoDataFoundException`. What is the most likely cause of this issue?

- A. The `Endpoint_URL__c` field is not accessible to the running user's profile
- B. The Apex class lacks the `with sharing` keyword
- C. The `Primary_Endpoint` record is not included in the deployment package
- D. The custom metadata type is not activated in the target environment

Answer: C

Explanation: Custom metadata records are treated as metadata and must be explicitly included in deployment packages, unlike regular data records. If the `Primary_Endpoint` record is missing in the target environment, `getInstance('Primary_Endpoint')` returns null, causing a `NoDataFoundException` when attempting to access `Endpoint_URL__c`.

#### Question: 464

An administrator must ensure that a custom object's records are only editable during a specific business process stage, with changes tracked for compliance. What should the administrator configure?

- A. Build a permission set with a Flow for stage-based access
- B. Create a validation rule with an Apex class for auditing
- C. Configure dynamic forms with a workflow for tracking
- D. Use a record-triggered Flow with field history tracking

Answer: D

Explanation: A record-triggered Flow can restrict edits based on process stage, and field history tracking logs changes for compliance. Validation rules with Apex are complex, dynamic forms don't enforce access, and permission sets with Flows are less efficient.

#### Question: 465

An administrator configures a process builder to update case fields based on entitlement milestones. The

process fails for cases with expired milestones, citing a “null pointer” error. What should the administrator check?

- A. The entitlement template’s active status
- B. The case object’s field-level security settings
- C. The debug log for entitlement process errors
- D. The process builder’s logic for handling null milestone values

Answer: D

Explanation: A “null pointer” error in a process builder indicates the process attempts to access a null value, likely due to expired milestones. Checking the process builder’s logic for handling null milestone values ensures it accounts for such cases. Field-level security affects visibility, not null errors. Debug logs may confirm the error but not the fix. Entitlement status doesn’t directly cause null pointer issues.

### Question: 466

A global organization tracks employee certifications in a custom object, with a requirement to notify external training providers via API when certifications expire, but only for employees in specific regions. The process must be scalable and minimize API calls. What should the administrator implement?

- A. Scheduled Flow with HTTP Callout and regional filters
- B. Outbound Message with Workflow Rules and a middleware
- C. Apex Trigger with asynchronous @future callouts
- D. Process Builder with invocable Apex for API integration

Answer: A

Explanation: A Scheduled Flow can run daily, filter certifications expiring for employees in specific regions, and use an HTTP Callout action to notify external providers in bulk, minimizing API calls through aggregation. Outbound Messages are tied to Workflow Rules, which are outdated and less flexible for filtering. Apex Trigger with @future risks callout limits in high-volume scenarios. Process Builder is deprecated and less efficient for scheduled, bulk API operations compared to Flow.

### Question: 467

Northern Trail Outfitters uses a custom object Employee\_\_c with a Lookup to User. A Visualforce page displays employee details using:

Users report a System.SObjectException: SObject row was retrieved via SOQL without querying the requested field. What should the administrator do?

- A. Use a formula field to display User\_\_r.Name
- B. Replace outputField with outputText
- C. Grant Read access to the User object for the user's profile
- D. Add User\_\_r.Name to the SOQL query in a controller extension

Answer: D

Explanation: The error occurs because Visualforce doesn't automatically query related fields like User\_\_r.Name when using standardController. A controller extension with a SOQL query (e.g., SELECT Id, User\_\_r.Name FROM Employee\_\_c WHERE Id = :recordId) ensures the field is retrieved.

**Question: 468**

Northern Trail Outfitters has a requirement to notify an external system when a Contract is activated, but only if related Opportunities meet specific revenue thresholds. The notification must include dynamic data and handle failures. What is the best solution?

- A. Configure a Platform Event with an Apex subscriber
- B. Use a Flow with fault handling
- C. Write a Batch Apex job with callouts
- D. Set up an Outbound Message with a callback

Answer: A

Explanation: A Platform Event with an Apex subscriber can send dynamic notifications, evaluate thresholds, and handle failures reliably. Flows are less robust for integrations, Batch Apex is not real-time, and Outbound Messages lack advanced error handling.

**Question: 469**

An administrator at Universal Containers needs to calculate a Risk\_Score\_\_c on Opportunity based on weighted criteria:

- Amount > \$100,000: +40
- Probability < 50%: +20

- Industry = 'Finance': +30

The administrator uses a formula field:

```
IF(Amount > 100000, 40, 0) +  
IF(Probability < 50, 20, 0) +  
IF(Industry = 'Finance', 30, 0)
```

During testing, Risk\_Score\_\_c is null for some records. What should the administrator do?

- A. Use a Flow to calculate the score
- B. Check for null values in the formula inputs
- C. Create an Apex trigger to set the score
- D. Add default values to Opportunity fields

Answer: B

Explanation: Formula fields return null if any input (Amount, Probability, Industry) is null, causing unexpected results. Checking and handling nulls in the formula (e.g., `IF(ISBLANK(Amount), 0, IF(Amount > 100000, 40, 0))`) ensures accurate calculations. A Flow or Apex trigger is overkill for a simple calculation. Default values may not align with business needs. Addressing nulls in the formula fixes the issue efficiently.

### Question: 470

Universal Containers uses a private sharing model with complex criteria-based sharing for opportunities. The administrator must ensure sales reps only see relevant records. Which three configurations are necessary? Choose 3 answers

- A. Criteria-Based Sharing Rules
- B. Role Hierarchy for team access
- C. Apex Sharing for dynamic logic
- D. Manual Sharing for exceptions
- E. Permission Sets for field access
- F. Profiles for object restrictions

Answer: A, C, E

Explanation: Criteria-Based Sharing Rules grant opportunity access based on record attributes, aligning with the private model. Apex Sharing handles complex, dynamic sharing logic beyond declarative rules. Permission Sets control field-level access, ensuring reps see only relevant data. Role Hierarchy is less flexible for criteria-based needs. Manual Sharing doesn't scale. Profiles are too rigid for dynamic field access.

Killexams.com is a leading online platform specializing in high-quality certification exam preparation. Offering a robust suite of tools, including MCQs, practice tests, and advanced test engines, Killexams.com empowers candidates to excel in their certification exams. Discover the key features that make Killexams.com the go-to choice for exam success.



## Exam Questions:

Killexams.com provides exam questions that are experienced in test centers. These questions are updated regularly to ensure they are up-to-date and relevant to the latest exam syllabus. By studying these questions, candidates can familiarize themselves with the content and format of the real exam.

## Exam MCQs:

Killexams.com offers exam MCQs in PDF format. These questions contain a comprehensive collection of questions and answers that cover the exam topics. By using these MCQs, candidate can enhance their knowledge and improve their chances of success in the certification exam.

## Practice Test:

Killexams.com provides practice test through their desktop test engine and online test engine. These practice tests simulate the real exam environment and help candidates assess their readiness for the actual exam. The practice test cover a wide range of questions and enable candidates to identify their strengths and weaknesses.

## Guaranteed Success:

Killexams.com offers a success guarantee with the exam MCQs. Killexams claim that by using this materials, candidates will pass their exams on the first attempt or they will get refund for the purchase price. This guarantee provides assurance and confidence to individuals preparing for certification exam.

## Updated Contents:

Killexams.com regularly updates its question bank of MCQs to ensure that they are current and reflect the latest changes in the exam syllabus. This helps candidates stay up-to-date with the exam content and increases their chances of success.