Up-to-date Practice Test with Latest Questions and Answers covering latest syllabus and topics of the exam. Makes you ready to face actual exam.

KILL EXAMS

Servicenow-CIS-CSM Practice Questions
Servicenow-CIS-CSM Practice Test
Servicenow-CIS-CSM Practice Exam
Servicenow-CIS-CSM Exam Questions
Servicenow-CIS-CSM Study Guide

killexams.com

ServiceNow

Servicenow-CIS-CSM

Certified Implementation Specialist - Customer Service Management (CIS-CSM)

ORDER FULL VERSION

https://killexams.com/pass4sure/exam-detail/Servicenow-CIS-CSM

Question: 522

Scenario: During the Execute phase, a technical consultant needs to configure a business rule to prevent case closure if a related task is open. Which script achieves this?

A. (function executeRule(current, previous) { var gr = new GlideRecord('task'); gr.addQuery('parent', current.sys_id); gr.addQuery('state', '!=', 'closed'); gr.query(); if (gr.hasNext()) { current.setAbortAction(true); } })(current, previous);
B. (function executeRule(current, previous) { var gr = new GlideRecord('task'); gr.addQuery('parent', current.sys_id); gr.addQuery('state', 'open'); gr.query(); if (gr.next()) { current.setAbortAction(true); } })(current, previous);
C. (function executeRule(current, previous) { var gr = new GlideRecord('task'); gr.addQuery('case', current.sys_id); gr.addQuery('state', '!=', 'closed'); gr.query(); if (gr.hasNext()) { gs.addErrorMessage('Cannot close case'); } })(current, previous);
D. (function executeRule(current, previous) { var gr = new GlideRecord('task'); gr.addQuery('parent', current.sys_id); gr.addQuery('state', 'open'); gr.query(); if (gr.hasNext()) { current.state = 'open'; } })(current, previous);

Answer: A

Explanation: The correct script uses GlideRecord to check for open tasks (state != closed) linked to the case via the parent field and aborts the closure with setAbortAction(true) if any are found. The other options are incorrect: state = open is too restrictive, gs.addErrorMessage does not prevent closure, and resetting the state is not appropriate.

Question: 523

A company is importing knowledge articles from an external CMS. The articles must retain their original publication dates. Which configuration ensures this?

A. Use a Scripted REST API to update dates after import
B. Use a business rule to set the publication date post-import
C. Modify the CMS to align dates with ServiceNow format
D. Map the source publication date to sys_created_on in a Transform Map

Answer: D

Explanation: Mapping the source publication date to sys_created_on in a Transform Map ensures articles retain their original dates during import. A business rule post-import is less efficient, modifying the CMS is impractical, and a Scripted REST API is unnecessary for bulk imports.

Question: 524

In a Customer Service Management (CSM) implementation, a company wants to configure the Customer Service Portal (/csp) to allow self-registration for contacts using unique registration codes per account.

The registration code must be validated before creating a new contact record, and only users with the 'sn_customerservice.customer' role should be able to self-register. Which configuration steps are required to achieve this?

A. Activate the 'com.glideapp.servicecatalog.security' plugin to enable registration code validation
B. Configure the 'sn_customerservice.self_registration' system property to true and set the 'registration_code_field' to 'u_registration_code'
C. Create a business rule on the 'sn_customerservice_contact' table to validate the registration code against the 'sn_customerservice_account' table
D. Modify the 'Self-Registration' widget in the Service Portal to include a client script that checks the 'sn_customerservice.customer' role

Answer: B,C

Explanation: To enable self-registration with unique registration codes in the Customer Service Portal, the 'sn_customerservice.self_registration' system property must be set to true, and the 'registration_code_field' must be configured to point to a custom field (e.g., 'u_registration_code') on the 'sn_customerservice_account' table to store and validate the code. A business rule on the 'sn_customerservice_contact' table is needed to validate the entered registration code against the account's stored code during registration. The 'com.glideapp.servicecatalog.security' plugin is unrelated to self-registration (it pertains to Service Catalog security), and modifying the 'Self-Registration' widget to check roles is unnecessary since role-based access is handled by the portal’s ACLs.

Question: 525

A Form Header in the CSM Configurable Workspace displays a case’s account.name and a custom field u_time_to_resolve (in hours). Which script calculates u_time_to_resolve based on the case’s SLA?

A. var sla = new GlideRecord('sla'); sla.addQuery('task', current.sys_id); sla.query(); if (sla.next()) { return gs.dateDiff(sla.end_time, sla.start_time, false); }
B. var sla = new GlideRecord('task_sla'); sla.addQuery('task', current.sys_id); sla.query(); if (sla.next()) { return (sla.end_time - sla.start_time) / 3600; }
C. var sla = new GlideRecord('task_sla'); sla.addQuery('task', current.sys_id); sla.query(); if (sla.next()) { return Math.floor(gs.dateDiff(sla.end_time, sla.start_time, true) / 3600); }
D. var sla = new GlideRecord('task_sla'); sla.addQuery('task', current.sys_id); sla.query(); if (sla.next()) { return Math.floor((sla.end_time - gs.nowDateTime()) / 3600); }

Answer: C

Explanation: The task_sla table stores SLA records, and gs.dateDiff(sla.end_time, sla.start_time, true) calculates the SLA duration in seconds, converted to hours with Math.floor().

Question: 526

In a Customer Service Management (CSM) implementation, a Quick Action in Workspace Chat is configured to resolve a case with the command /r "Resolution notes". A new requirement demands that this command also updates the case's customer satisfaction score to 5 and sets the state to "Closed." Which script modification in the Quick Action definition achieves this?

A. current.state = 'Closed'; current.satisfaction_score = 5; current.update();
B. current.setValue('state', 'Closed'); current.setValue('satisfaction_score', 5); current.update();
C. current.state = 3; current.satisfaction_score = 5; current.update();
D. current.setValue('state', 3); current.setValue('satisfaction_score', '5'); current.update();

Answer: B

Explanation: In ServiceNow, the correct syntax for updating fields programmatically uses setValue to ensure proper field updates, especially for string or choice list fields like state and satisfaction_score. The state field for a case (e.g., sn_customerservice_case) typically uses string values like "Closed" rather than numeric codes in modern releases (C, D are incorrect). Using current.state directly may work but is less reliable for choice fields, and D incorrectly sets satisfaction_score as a string. Thus, B is the most robust solution.

Question: 527

In a Customer Service Management (CSM) implementation, a company wants to configure the Customer Service Portal (/csp) to allow self-registration for contacts using unique registration codes per account. The registration code must be validated before creating a new contact record, and only users with the 'sn_customerservice.customer' role should be able to self-register. Which configuration steps are required?

A. Activate the 'com.glideapp.servicecatalog.security' plugin to enable registration code validation
B. Configure the 'sn_customerservice.self_registration' system property to true and set the 'registration_code_field' to 'u_registration_code'
C. Create a business rule on the 'sn_customerservice_contact' table to validate the registration code against the 'sn_customerservice_account' table
D. Modify the 'Self-Registration' widget in the Service Portal to include a client script that checks the 'sn_customerservice.customer' role

Answer: B,C

Explanation: To enable self-registration with unique registration codes in the Customer Service Portal, the 'sn_customerservice.self_registration' system property must be set to true, and the 'registration_code_field' must be configured to point to a custom field (e.g., 'u_registration_code') on the 'sn_customerservice_account' table to store and validate the code. A business rule on the 'sn_customerservice_contact' table is needed to validate the entered registration code against the account's stored code during registration. The 'com.glideapp.servicecatalog.security' plugin is unrelated to self-registration (it pertains to Service Catalog security), and modifying the 'Self-Registration' widget to check

roles is unnecessary since role-based access is handled by the portal’s ACLs.

Question: 528

A customer reports that cases are not routing correctly to agent groups in AWA. The issue is due to misconfigured matching rules. Which table should you check to resolve this?

A. awa_assignment_rule
B. awa_condition
C. sn_customerservice_case
D. sys_user_group

Answer: B

Explanation: AWA matching rules are configured in the awa_condition table, which defines how cases are routed to agent groups based on conditions. The awa_assignment_rule table defines assignment logic, not matching rules. The case table stores case data, and sys_user_group manages groups, not routing rules.

Question: 529

A company wants to route cases to agents based on complex Matching Rules in the Assignment Workbench, prioritizing agents with the sn_customerservice_agent role, fluency in the customer’s language (stored in user.language), and a workload below 10 open cases. Which conditions must be included in the Matching Rules setup?

A. Agent has role 'sn_customerservice_agent'
B. Agent.language matches case.customer.language
C. Agent’s open case count < 10
D. Agent’s skills match case.u_category

Answer: A,B,C

Explanation: The Matching Rules must filter agents by the sn_customerservice_agent role, match their language to the customer’s language (via user.language and case.customer.language), and ensure their workload (open case count) is below 10. The Assignment Workbench supports these conditions natively. Matching skills to case.u_category is not mentioned in the requirement, making choice D irrelevant.

Question: 530

A company uses entitlements to limit case creation to 5 cases per month per account. Which script in a Business Rule enforces this limit?

A. var gr = new GlideRecord('sn_customerservice_case'); gr.addQuery('account', current.account); gr.addQuery('sys_created_on', '>=', gs.daysAgo(30)); gr.query(); if(gr.getRowCount() >= 5) current.setAbortAction(true);
B. var gr = new GlideRecord('sn_customerservice_case'); gr.addQuery('account', current.account); gr.addQuery('created_on', '>=', gs.nowDateTime()); gr.query(); if(gr.getRowCount() >= 5) current.setAbortAction(true);
C. var gr = new GlideRecord('sn_customerservice_case'); gr.addQuery('account', current.account); gr.query(); if(gr.getRowCount() >= 5) current.setAbortAction(true);
D. var gr = new GlideRecord('sn_customerservice_case'); gr.addQuery('account', current.account); gr.addQuery('sys_created_on', '>=', gs.beginningOfThisMonth()); gr.query(); if(gr.getRowCount() >= 5) current.setAbortAction(true);

Answer: D

Explanation: To enforce a 5-case-per-month limit, the Business Rule queries the sn_customerservice_case table for cases created this month (sys_created_on >= gs.beginningOfThisMonth()) for the same account. If the count is 5 or more, setAbortAction(true) prevents case creation.

Question: 531

A global retailer expects its CSM implementation to handle 30,000 daily cases. To ensure scalability, the team is configuring asynchronous processing for case updates. Which script include function should be used?

A. CaseProcessor.asyncUpdate(case)
B. AsyncCaseManager.process(case, updates)
C. GlideRecordAsync.update(case, fields)
D. BackgroundJob.runCaseUpdate(case)

Answer: C

Explanation: The GlideRecordAsync.update function is ServiceNow’s recommended method for asynchronous database updates, ensuring scalability by offloading case updates to background processes. Other options are not standard ServiceNow functions for asynchronous processing, making them less suitable for high-volume case updates.

Question: 532

A Quick Action /priority sets a case’s priority to "Critical." A new requirement mandates that it also creates a related task for a senior agent to review the case. Which script achieves this?

A. current.priority = 'Critical'; var task = new GlideRecord('task'); task.initialize(); task.short_description

= 'Review Case'; task.insert();
B. current.setValue('priority', 'Critical'); var task = new GlideRecordSecure('task'); task.initialize(); task.short_description = 'Review Case'; task.insert();
C. current.priority = 1; var task = new GlideRecordSecure('task'); task.initialize(); task.short_description = 'Review Case'; task.insert();
D. current.setValue('priority', 1); var task = new GlideRecord('task'); task.initialize(); task.short_description = 'Review Case'; task.insert();

Answer: B

Explanation: The script should use setValue for the priority field (string value "Critical") and GlideRecordSecure for creating a related task to comply with Washington release security standards.

Question: 533

Scenario: A ServiceNow administrator needs to configure the CSM Portal to allow customers to submit cases directly from a knowledge article page. Which configuration enables this functionality?

A. Create a UI Action on the knowledge article form to trigger case creation
B. Enable the com.sn_customerservice_case plugin and configure a Record Producer
C. Set up a Contextual Search action to link articles to case submission
D. Use a business rule to redirect users to the case creation form

Answer: B

Explanation: To allow customers to submit cases directly from a knowledge article page in the CSM Portal, the com.sn_customerservice_case plugin must be activated to enable case management functionality. A Record Producer should be configured to provide a simplified interface for case submission, accessible directly from the portal's knowledge article page.

Question: 534

After upgrading to the Washington release, a custom Quick Action /assign that assigns cases to a specific group fails due to a change in the Workspace Chat API. The original script used GlideRecord to query the sn_customerservice_case table. What should be done to restore functionality?

A. Update the Quick Action to use a Flow Designer flow instead
B. Convert the Quick Action to a UI Action and retest
C. Revert the instance to the previous release and debug
D. Replace GlideRecord with GlideRecordSecure in the Quick Action script

Answer: D

Explanation: The Washington release introduced stricter security for scripts, requiring GlideRecordSecure instead of GlideRecord to ensure Feature Parity and compliance with security standards. Converting to a UI Action or using Flow Designer changes the functionality and is unnecessary. Reverting the instance is not a viable solution for maintaining Feature Parity.

Question: 535

A company is upgrading its ServiceNow instance from Vancouver to Washington DC. A custom application in the CSM module uses a scripted REST API that directly modifies the sn_customerservice_case table. To ensure upgradability, which two practices should the team follow?

A. Encapsulate the API logic in a scoped application
B. Use global scripts to maintain compatibility
C. Leverage the ServiceNow REST API Explorer for testing
D. Store custom logic in the sys_script table

Answer: A, C

Explanation: To ensure upgradability, encapsulating custom logic in a scoped application isolates it from platform changes, reducing conflicts during upgrades. Using the REST API Explorer for testing ensures compatibility with ServiceNow's API standards. Storing logic in the sys_script table or using global scripts increases the risk of upgrade conflicts, as these are not isolated and may be overwritten or cause issues with new platform features.

Question: 536

During an upgrade from Utah to Vancouver, which step ensures that the CSM mobile app's offline data sync frequency remains unchanged?

A. Export and import the sys_mobile_offline_sync_config table
B. Update the sys_mobile_sync_frequency script include
C. Reconfigure the Mobile App Configuration sync settings
D. Set the sys_offline_sync_frequency property to the previous value

Answer: A

Explanation: To ensure the CSM mobile app's offline data sync frequency remains unchanged during an upgrade from Utah to Vancouver, you should export and import the sys_mobile_offline_sync_config table, which stores sync frequency settings. There is no sys_mobile_sync_frequency script include or sys_offline_sync_frequency property, and reconfiguring the Mobile App Configuration is not the standard approach for preserving sync settings.

Question: 537

Scenario: A retail organization is implementing CSM to handle customer complaints. The Now Create methodology recommends using the Customer Service Playbook. The team needs to configure a dynamic SLA based on case priority and customer tier. Which script include function should be used to calculate the SLA duration dynamically?

A. SLACalculator.getDuration(priority, tier)
B. calculateSLADuration(casePriority, tierLevel)
C. getDynamicSLADuration(priority, customerTier)
D. DynamicSLA.computeDuration(case, tier)

Answer: A

Explanation: The SLACalculator.getDuration function in ServiceNow's Customer Service Playbook is designed to calculate dynamic SLA durations based on parameters like case priority and customer tier. This function aligns with Now Create's best practices for SLA management, ensuring maintainable and scalable configurations. Other options are not standard ServiceNow functions for SLA calculations.

Question: 538

You need to configure a Service Catalog item to trigger a notification when a case is assigned to a specific group. Which approach is best?

A. Create a notification triggered by a business rule
B. Modify the catalog item's script include
C. Write a client script to trigger the notification
D. Use Flow Designer to send the notification

Answer: D

Explanation: Flow Designer is the best practice for triggering notifications based on case assignment, as it provides a scalable, low-code solution. Business rules are less flexible. Client scripts are inappropriate for server-side notifications. Modifying script includes is unnecessary.

Question: 539

A client requires a custom integration between ServiceNow CSM and a third-party CRM system to synchronize customer data in real time, ensuring upgradability and scalability. Which approach best preserves the platform's maintainability while meeting the requirement?

A. Develop a Flow Designer flow using IntegrationHub with REST spokes and error handling
B. Create a custom Script Include with REST API calls and schedule it using a Scheduled Job
C. Use a Business Rule to trigger SOAP-based data sync on every record update
D. Implement a custom UI Page with client-side scripting to push data to the CRM

Answer: A

Explanation: Using Flow Designer with IntegrationHub (REST spokes) is the recommended approach for custom integrations in ServiceNow as it leverages out-of-the-box tools, supports error handling, and ensures compatibility with upgrades. Scheduled Jobs with Script Includes increase maintenance overhead, Business Rules with SOAP are less scalable due to synchronous processing, and UI Pages with client-side scripting are not suitable for server-side integrations.

Question: 540

A company configures a service contract with entitlements tied to Predictive Intelligence outputs. Which field in the sn_customerservice_entitlement table is used to store the Predictive Risk Score?

A. predictive_score
B. risk_score
C. entitlement_risk
D. predictive_risk_score

Answer: D

Explanation: The predictive_risk_score field is used to store Predictive Intelligence outputs in the sn_customerservice_entitlement table. Other options are not standard fields.

Question: 541

A company restricts the "sn_customerservice_agent" and "sn_customerservice_consumer_agent" roles from being assigned together. Which configuration enforces this?

A. Set a UI policy on "sys_user_has_role" to block role assignments
B. Create a business rule to check role assignments
C. Define a role conflict policy in "sys_user_role_conflict"
D. Use an ACL to restrict role assignments

Answer: C

Explanation: A role conflict policy in the "sys_user_role_conflict" table prevents the two roles from being assigned together. A business rule is overly complex, a UI policy cannot enforce role conflicts, and an

ACL is not suitable for role assignment restrictions.

Question: 542

During the Deploy phase exit gate, the team identifies that the CSM portal's custom widgets are causing performance issues under high user load. Which checkpoint should be prioritized to address this before go-live?

A. Performance Testing and Optimization
B. Security Configuration Validation
C. Data Integrity Check
D. Integration Testing Completion

Answer: A

Explanation: The Performance Testing and Optimization checkpoint during the Deploy phase exit gate ensures that custom components, such as portal widgets, are optimized for high user loads. This is critical for scalability and user experience in the CSM portal. Other checkpoints like Security Configuration Validation, Data Integrity Check, and Integration Testing focus on different aspects and are less relevant to widget performance.

Question: 543

An organization needs to restrict case updates to agents with the 'sn_customerservice_manager' role for cases linked to high-priority accounts. Which configuration enforces this?

A. Set up a Data Policy to restrict case updates
B. Use a UI Policy to disable updates for non-managers
C. Create a Business Rule to check user roles and account priority
D. Configure an ACL on 'sn_customerservice_case' with a condition for account priority

Answer: D

Explanation: An Access Control List (ACL) on the 'sn_customerservice_case' table can restrict updates by requiring the 'sn_customerservice_manager' role and checking the account's priority (e.g., 'account.priority' = 'High'). ACLs enforce server-side security, ensuring robust access control. UI Policies are client-side and less secure. Business Rules are for data operations, not access control. Data Policies enforce data validation, not role-based access.

KILLEXAMS.COM

Killexams.com is a leading online platform specializing in high-quality certification exam preparation. Offering a robust suite of tools, including Exam Questions, 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.

Practice Exam Questions Based on Current Exam Objectives

Killexams.com provides practice exam questions aligned with the latest official exam objectives and latest syllabus. Our content is reviewed and updated regularly to reflect recent changes announced by certification vendors. By studying these practice questions, candidates will cover the structure, difficulty level, and topics of the actual exam, helping them prepare more effectively and efficiently.

Comprehensive Practice Exams (PDF Format)

Killexams.com offers multiple-choice questions (MCQs) in easy-to-read PDF format, covering all major domains of the exam. Each PDF contains a structured collection of practice questions and verified answers designed to support focused study. These MCQs help candidates reinforce key concepts, identify knowledge gaps, and improve exam readiness through consistent practice.

Realistic Practice Tests (Online Test Engine & Desktop Test Engine)

To support hands-on preparation, Killexams.com provides practice tests through both an Online Test Engine and a Desktop Test Engine. These tools are designed to simulate a real exam environment, allowing candidates to practice under exam-like conditions, with latest syllbus and topics of the exam. Performance tracking, test history, and result analysis help users evaluate their progress and focus on areas that need improvement.

Risk-Free Purchase Policy

Killexams.com follows a transparent and customer-friendly purchase policy. If users are not satisfied with the study materials, they may request assistance or a refund in accordance with our published terms and conditions. This policy reflects our commitment to customer satisfaction, fairness, and confidence in our preparation resources.

Regularly Updated Content

Our practice question bank is reviewed and updated on an ongoing basis to stay aligned with the latest exam outlines and vendor updates. This ensures candidates are studying up-to-date, relevant material, and preparing with content that reflects current exam expectations, helping them stay confident and well-prepared.