



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.



ServiceNow-CAD Practice Questions
ServiceNow-CAD Practice Test
ServiceNow-CAD Practice Exam
ServiceNow-CAD Exam Questions
ServiceNow-CAD Study Guide



killexams.com

ServiceNow

ServiceNow-CAD

ServiceNow Certified Application Developer (CAD)

ORDER FULL VERSION

<https://killexams.com/pass4sure/exam-detail/ServiceNow-CAD>



Question: 1226

When configuring a Data Source for a CSV import, which option determines how records will be identified as duplicates?

- A. Primary Key
- B. Unique Identifier
- C. Coalesce Field
- D. Record ID

Answer: C

Explanation: The Coalesce Field in a Data Source configuration determines how records will be identified as duplicates. If the coalesce field matches an existing record, that record will be updated instead of creating a new one.

Question: 1227

Responsive form in Service Portal for travel requests validates attachments size <5MB with File API, client script onChange="if (file.size > 5*1024*1024) rejectFile();". Theme .upload-zone { border: 2px dashed var(--upload-border); }. On iOS, drag-drop fails, reverting to click. Mobile form .col-sm-6 jumps on reject. What API polyfill and theme stabilize?

- A. Script: `if (window.FileReader) { reader = new FileReader(); reader.onload = e => { if (e.total > 5242880) { alert('Too large'); return; } }; } else { /* polyfill */ } and theme CSS .col-sm-6 { transition: none; min-height: 100px; } @media (max-width: 576px) { .upload-zone { border-style: solid; } }.`
- B. Use ng-file-upload, and theme SCSS \$file-reject: box-shadow: inset 0 0 0 3px red;.
- C. Server size check, and theme variable --jump-prevent: position: relative;.
- D. Dropzone.js lib, and media query .form-row { align-items: stretch; }.

Answer: A

Explanation: The FileReader check with polyfill handles iOS drag-drop fallback, validating size <5MB accurately. Theme .col-sm-6 { transition: none; min-height: 100px; } prevents jumps on reject, with @media mobile adjusting .upload-zone border for touch, ensuring responsive stability in travel form.

Question: 1228

In ServiceNow, which property must be configured to enable the use of Git branches for

application development?

- A. glide.git.versioning.enabled
- B. glide.git.integration.enabled
- C. glide.git.repository.enabled
- D. glide.git.branching.enabled

Answer: D

Explanation: The property `glide.git.branching.enabled` must be set to true to allow developers to use Git branches for managing different versions of their applications.

Question: 1229

A developer is implementing a GlideSystem method to log security events. Which method should be used to ensure that the event is logged correctly with the appropriate level of severity?

- A. gs.info("message")
- B. gs.error("message")
- C. gs.logEvent("event_name", "message")
- D. gs.warning("message")

Answer: C

Explanation: The method `gs.logEvent("event_name", "message")` is used to log security events with a specified event name, allowing for better tracking and management of security-related activities.

Question: 1230

For a Script Include utility in a scoped app, the getRelatedMetrics() function uses GlideRecord on metric_instance with addQuery('table', 'incident').addQuery('field', 'state'), grouping by sys_ids via addAggregate('COUNT', 'value'), then returns a JSON object for client consumption. Called from a beforeQuery Business Rule to filter low-metric incidents. In Vancouver release, it hits query limits on large tables. What performance tweak uses indexed fields?

- A. Add gr.setLimit(1000); and gs.cache() the results with 1-hour TTL for repeated calls.
- B. Switch to GlideAggregate exclusively, adding addGroupBy('record.sys_id') before query, ensuring indexes on state and sys_id.
- C. Integrate with Performance Analytics for pre-aggregated metrics, querying

pa_indicators via REST API.

D. Use `gs.query()` with raw SQL for custom grouping, bypassing `GlideRecord` abstractions.

Answer: B

Explanation: `GlideAggregate` with `addGroupBy('record.sys_id')` efficiently groups metrics without full table scans, assuming indexes on `metric_instance.state` and `record(sys_id)`. Code:

```
var ga = new GlideAggregate('metric_instance'); ga.addQuery('table', 'incident'); ga.addQuery('field', 'state'); ga.addAggregate('COUNT'); ga.addGroupBy('record.sys_id'); ga.query(); var metrics = {}; while(ga.next()) metrics[ga.record.sys_id] = ga.getAggregate('COUNT'); return JSON.stringify(metrics);
```

 This resolves limits in `beforeQuery` rules for scaled environments.

Question: 1231

Washington scenario: Multi-US merge for "Integration Hub Extensions," parent US "IH-Parent" references child "IH-Child" with `sys_update_set.parent='ih_parent_sys_id.'` Merge fails: "Orphaned child: no parent link." What query and update script repairs?

- A.

```
var child = new GlideRecord('sys_update_set'); child.addQuery('name', 'IH-Child'); child.query(); if (child.next()) { child.parent = 'ih_parent_sys_id'; child.update(); } gs.validateMerge('parent_id');
```
- B. Delete child.
- C. Manual link.
- D. Query child US; set parent; `gs.mergeUS('IH-Parent', 'children')`.

Answer: A

Explanation: Orphaned children break merges. The `GlideRecord` links them explicitly, with `validateMerge` ensuring integrity before proceeding in multi-US hierarchies.

Question: 1232

In a form, how can a developer dynamically change the visibility of a field based on another field's value?

- A. Using a UI Policy
- B. Using a Business Rule
- C. Using a Data Policy
- D. Using a Client Script

Answer: A

Explanation: UI Policies allow developers to dynamically change the visibility of fields based on the value of other fields, enhancing user experience.

Question: 1233

What is the result of setting a field's "Visible" property to false in the Form Layout?

- A. The field is still accessible via scripts but not displayed on the form.
- B. The field is removed from the database.
- C. The field is hidden from all users, regardless of roles.
- D. The field can still be accessed through reports.

Answer: A

Explanation: Setting a field's "Visible" property to false hides it from the form but does not remove it from the database, allowing it to be accessed programmatically.

Question: 1234

Integrating with paginated API /data?cursor=next, where cursor is opaque string, but script assumes numeric page. To handle cursor param, loop until null cursor in response, accumulate results >1000 rows with memory check, and abort if >10k, what code?

- A. `r.setQueryParameterNoEscape('cursor', cursor);`
- B. `for (var page=1; page<10; page++) { r.setQueryParameter('page', page); }`
- C. `var total = 0; while (total < 1000) { /* pull */ }`
- D. `var results = []; var cursor = ""; do { var r = new sn_ws.RESTMessageV2('Data_Pull', 'GET'); if (cursor) r.setQueryParameter('cursor', cursor); var resp = r.execute(); var body = JSON.parse(resp.getBody()); results = results.concat(body.data); cursor = body.next_cursor; if (results.length > 10000) { gs.error('Exceeded limit'); break; } if (gs.getMemoryUsage() > 0.8) { gs.sleep(5000); } } while (cursor); /* process results */`

Answer: D

Explanation: Looping with cursor param until null, concatenating data arrays, checking length for abort and memory usage for pauses prevents overload, handling opaque cursors correctly for full pagination.

Question: 1235

Form 'Software License': sections 'License Info' (product ref, seats integer), 'Usage' embedded list (user ref, filter 'license=^EQ'), 'Compliance' (ratio calc seats/used). UI Policy: if ratio <0.8, show 'Overusage Alert' mandatory action plan. onChange product update seats default GlideAjax.

- A. No policy.
- B. No embedded.
- C. Server calc.
- D. Layout embedded filter; calc dictionary; policy ratio<0.8 show/mandatory; onChange GlideAjax set seats.

Answer: D

Explanation: Layout with embedded usage list filtered; dictionary calc 'used = count users, ratio=seats/used'. Policy condition ratio<0.8 shows alert section mandatory. onChange GlideAjax to product include sets default seats, for license management in Yokohama.

Question: 1236

In ServiceNow, which of the following methods can be used to create a new record in a table?

- A. GlideRecord.create()
- B. GlideRecord.newRecord()
- C. GlideRecord.add()
- D. GlideRecord.insert()

Answer: D

Explanation: The `insert()` method is used to create a new record in the specified table, while `create()` and `newRecord()` are not valid GlideRecord methods.

Question: 1237

In a high-availability setup, a Scheduled Job executes every 15 minutes to process pending 'u_batch_import' records, uses GlideMultipleUpdate for bulk updates on up to 10,000 records matching 'status=pending' and 'import_date < gs.minutesAgoStart(30)', calculates checksums via a custom Script Include for each batch, and fires 'batch.processed' event per successful batch with parameters for row count and errors array. To prevent timeouts, it must chunk queries in 1,000-record batches. What is the advanced script for this job?

```

A. var offset = 0; do { var gmu = new GlideMultipleUpdate('u_batch_import');
gmu.addQuery('status', 'pending'); gmu.addQuery('import_date', '<',
gs.minutesAgoStart(30)); gmu.setLimit(1000, offset); var util = new
BatchChecksumUtil(); gmu.executeScript('if (current.status == "pending") {
current.checksum = util.compute(current.file_data); current.status = "processed"; }');
offset += 1000; } while (offset < 10000); gs.eventQueue('batch.processed', gmu, offset,
['']);
B. var gr = new GlideRecord('u_batch_import');
gr.addEncodedQuery('status=pending^import_dateC. var chunk = 0; while (true) { var gr
= new GlideRecord('u_batch_import'); gr.addQuery('status', 'pending');
gr.addQuery('import_date', '<', gs.minutesAgoStart(30)); gr.setLimit(1000, chunk *
1000); gr.query(); if (gr.getRowCount() == 0) break; var errors = [];
gr.forEach(function(record) { var chk = new
BatchChecksumUtil().compute(record.file_data); if (chk) { record.status = 'processed';
record.checksum = chk; record.update(); } else { errors.push(record.number); } });
chunk++; } gs.eventQueue('batch.processed', null, chunk * 1000, errors.join(','));
D. var offset = 0; var batchSize = 1000; var processed = 0; var errors = []; do { var gr =
new GlideRecord('u_batch_import'); gr.addQuery('status', 'pending');
gr.addQuery('import_date', '<', gs.minutesAgoStart(30)); gr.setLimit(batchSize);
gr.setWorkflow(false); gr.autoFlush(false); gr.queryWithLimit(offset, batchSize); if
(!gr.hasNext()) break; var gmu = new GlideMultipleUpdate(gr);
gmu.setInlineUpdates(true); while (gr.next()) { var checksum = new
BatchChecksumUtil().compute(gr.file_data); if (checksum) { gr.status = 'processed';
gr.checksum = checksum; } else { errors.push(gr.sys_id); } } gmu.execute(); processed
+= gr.getRowCount(); offset += batchSize; } while (offset < 10000); if (processed > 0) {
gs.eventQueue('batch.processed', null, processed.toString(), JSON.stringify(errors)); }

```

Answer: D

Explanation: The script in option A implements chunking with queryWithLimit(offset, batchSize) for precise pagination, uses GlideMultipleUpdate with autoFlush(false) and inline updates for bulk efficiency, computes checksums in the loop, accumulates errors, and queues the event with stringified array for complex parm2. It caps at 10,000 to prevent overload. Option B processes all at once risking timeouts, C uses forEach which is slower than GMU, and D misuses executeScript on GMU which doesn't support Script Includes directly.

Question: 1238

In "Vendor Risk Analyzer" scoped application, a pie chart module must slice data from "u_risk_assessments" by category, but incorporate global "sys_category" labels only

through a joined query filtered by app namespace to avoid label dilution. What server data script for the widget implements this join?

- A. `var join = new GlideRecord('u_risk_assessments'); join.addJoinQuery('sys_category', 'sys_id', 'u_category_ref'); join.addAggregate('COUNT_DISTINCT', 'sys_category.label'); join.query(); data.categories = join.getAggregates(); with filter namespace=x_vendor_.`
- B. Script: `var catGr = new GlideRecord('sys_category'); catGr.addQuery('sys_scope', gs.getCurrentScopeName()); catGr.query(); var data = []; for (var i=0; i< data.length; i++) { data[i] = new GlideRecord('u_risk_assessments'); data[i].addAggregate('COUNT'); data[i].addQuery('u_category', '!=', ''); data[i].addJoinQuery('sys_category', 'u_category', 'sys_id', 'label'); data[i].addQuery('sys_category.namespace', 'CONTAINS', 'x_vendor_'); data[i].query(); var slices = []; while(data[i].next()) { slices.push({label: data[i].label, value: data[i].getAggregate('COUNT')}); } data[i].slices = slices; }`
- D. Server script: `function getPieData() { var ns = gs.getScopeAbbreviation(); var q = new GlideEncodedQuery('u_risk_assessments', 'u_category!=empty'); q.addJoin('sys_category', {condition: 'namespaceLIKE' + ns}); return q.getAggregate('GROUP_CONCAT', 'u_category.label~COUNT'); } data.pie = getPieData();`

Answer: C

Explanation: Pie chart widgets in Vendor Risk Analyzer use `addJoinQuery()` with namespace filters on `sys_category` to aggregate counts by scoped labels only, producing clean slices. Washington DC's aggregate joins support this efficiently. Loops are slow for large data, and encoded queries don't join.

Question: 1239

A scoped feedback app's scheduled job runs weekly to analyze `x_feedback` records with `u_rating < 3`, uses `Utils` to sentiment-analyze text via integrated NLP property call, categorizes into themes using `switch` on keywords, aggregates counts per theme with `GlideAggregate`, generates bar chart HTML, attaches as PNG via `base64`, emails report if negative feedback `> 20%`, and queues `'feedback.improve'` event with top theme. What aggregation code produces the report?

- A. Job Script: `var util = new x_feedback.Utils(); util.analyzeWeekly();`
- B. Job Script: `var ga = new GlideAggregate('x_feedback'); ga.addQuery('u_rating', '<', 3); ga.addQuery('sys_created_on', '>=', gs.weeksAgoStart(1)); ga.addAggregate('COUNT'); ga.groupBy('u_theme'); ga.query(); var themes = {}; while(ga.next()) { themes[ga.u_theme.toString()] = ga.getAggregate('COUNT'); } var totalNeg =`

```

Object.values(themes).reduce((a,b)=>a+b,0); var topTheme =
Object.keys(themes).reduce((a,k)=> themes[k]>themes[a]?k:a); var html = '
'; var em = new GlideEmailOutbound();
em.setBodyHTML(html); if (totalNeg > 0.2 * totalNeg) { wait, if totalNeg >20% of all?
Assume all is totalNeg for neg only. Adjust: var allGa = new
GlideAggregate('x_feedback'); allGa.addQuery('sys_created_on', '>=',
gs.weeksAgoStart(1)); allGa.addAggregate('COUNT'); allGa.query(); var allCount =
allGa.getAggregate('COUNT'); if (totalNeg / allCount > 0.2) {
em.addAttachment(base64Chart, 'report.png', 'image/png'); gs.send(em);
gs.eventQueue('feedback.improve', topTheme, totalNeg); } };
C. Job Script: gr.query(); gr.deleteMultiple();
D. Job Script: gs.getMessage('report');

```

Answer: B

Explanation: The weekly job queries low-rated feedback from past week using `gs.weeksAgoStart(1)`, aggregates count per `u_theme` with `groupBy`, sums negative total, finds top theme with `reduce`, generates HTML with embedded `Chart.js` script for bar chart using JSON data, calculates percentage against all weekly feedback count from separate aggregate, attaches base64 PNG if `>20%`, sends email, and queues improvement event with theme and count, enabling data-driven insights.

Question: 1240

For a branded Service Portal navigation in an HR app, the theme uses CSS variables `--nav-link-hover: rgba(0,123,255,0.1)`; and requires RTL support for Arabic locales via `dir="rtl"`. The widget uses `spAriaFocusManager` for keyboard nav, but on RTL mobile, focus indicators shift left. The nav is `.sp-horizontal-nav` with flexbox. What theme and widget code enable RTL adaptation without breaking hover effects?

- A. Use AngularJS `$locale.id`, and theme variable `--rtl-margin: 0 1rem 0 0`; with media query for mobile RTL.
- B. Server script to set `c.data.rtl = gs.getSession().getLanguage() === 'ar'`; and theme SCSS `@mixin rtl { transform: scaleX(-1); }` for focus.
- C. Widget client: `spAriaFocusManager.init(c, '.sp-nav-link')`; and theme CSS `[dir="rtl"] .sp-horizontal-nav { flex-direction: row-reverse; } :focus { outline-offset: -2px; right: 0;`

};

D. CSS logical properties `margin-inline-start: 1rem;` and widget event `'keydown'` with `e.keyCode === 37` for RTL left arrow.

Answer: C

Explanation: Initializing `spAriaFocusManager` in the client controller ensures keyboard navigation, while `theme [dir="rtl"] .sp-horizontal-nav { flex-direction: row-reverse; }` reverses flexbox for RTL, adjusting `:focus { outline-offset: -2px; right: 0; }` to position indicators correctly on mobile. This preserves `--nav-link-hover` effects and supports Arabic locales in the HR app's branded navigation.

Question: 1241

In ServiceNow, which of the following is a best practice when handling API responses in server-side scripts?

- A. Ignore the response if the API call fails.
- B. Validate the response status code before processing the data.
- C. Always log the raw response for debugging purposes.
- D. Assume the response is always valid and proceed with processing.

Answer: B

Explanation: Validating the response status code before processing the data ensures that you only handle successful responses, preventing errors and ensuring data integrity.

Question: 1242

Scheduled script for `'u_backup_status'` queries `gr.addQuery('u_status', 'failed');` but ACL denies for non-admins. Secure background query?

- A. Script: `gs.getSession().impersonate('admin');` `gr.query();`
- B. Use `GlideAjax` to UI with admin role.
- C. ACL read: `answer = gs.getUserName() == 'system' || gs.getUser().hasRole('admin');`
- D. Run as user with backup role.

Answer: C

Explanation: Username check allows system queries without impersonation risks. This secures background, avoiding over-permissive. Job logs debug access.

Question: 1243

In "Sustainability Impact Calculator" scoped app, a combo module (line + bar) forecasts "u_emission_projections" against global "sys_forecast_baseline", but forecasts must blend scoped projections with baseline via a weighted linear regression scoped to app parameters. What regression script blends this?

- A. `function weightedBlend() { var scopeParam = gs.getNumberProperty('x_sustain_blend_weight', 0.5); var projGr = new GlideRecord('u_emission_projections'); projGr.query(); var results = []; while(projGr.next()) { var base = new GlideRecord('sys_forecast_baseline').get(projGr.u_year); results.push({year: projGr.u_year, value: projGr.u_emissions * scopeParam + base.value * (1 - scopeParam)}); } return results; } data.blend = weightedBlend();`
- B. `var regress = new GlideLinearRegression(); regress.addData('u_emission_projections', 'u_year', 'u_emissions', {scope: gs.getScopeId()}); regress.addBaseline('sys_forecast_baseline', 0.4); data.forecast = regress.predict(2026);`
- C. `Script: var x = new GlideArray(); x.loadQuery('u_emission_projections^sys_scope=' + gs.getCurrentScopeAbbreviation() + '^ORDERBYu_year'); var y = x.map(row => row.u_emissions); var baselineY = gs.getProperty('global.baseline_emissions'); var blendedY = y.map(v => (v + baselineY) / 2); data.combo = {x: x.getValues('u_year'), y_scoped: y, y_blend: blendedY};`
- D. `Combo chart, Server script: var scopedProj = []; var gr = new GlideRecord('u_emission_projections'); gr.addQuery('sys_scope', gs.getCurrentScopeName()); gr.orderBy('u_year'); gr.query(); while(gr.next()) { scopedProj.push({x: gr.u_year, y: gr.u_emissions}); } var baseline = new GlideRecord('sys_forecast_baseline'); baseline.get('default'); var weight = 0.6; // scoped weight var blended = scopedProj.map(p => ({x: p.x, y: (p.y * weight) + (baseline.value * (1 - weight))})); data.line = blended; data.bar = scopedProj;`

Answer: A

Explanation: Combo modules in Sustainability Impact Calculator use weighted averages in a function blending scoped projections with global baselines via `getNumberProperty()` for tunable scope param, producing arrays for line/bar. Xanadu's properties support dynamic weights. Maps assume fixed baseline, and regressions require ML plugins.

Question: 1244

Which of the following is the correct way to reference a parameter passed to an event in a script?

- A. `current.paramName`
- B. `event.paramName`
- C. `gs.event.paramName`
- D. `gs.getEventParameter('paramName')`

Answer: B

Explanation: In ServiceNow, parameters passed to an event can be referenced using the `'event.paramName'` syntax. This allows access to the data associated with the event.

Question: 1245

When creating a report widget for a dashboard, which of the following settings must be configured to ensure the report is visible to users?

- A. Configure a data policy
- B. Assign the report to a specific role
- C. Set the report to public
- D. Set the report as a favorite

Answer: C

Explanation: Setting the report to public ensures that all users with access to the dashboard can view the report, regardless of their roles.

Question: 1246

What is the correct sequence of steps to create a new field in an existing table?

- A. Use the Dictionary to create the field and then add it to the form
- B. Modify the form layout, add the field, and then access the Table Configuration
- C. Create a new form, add the field, and then save the changes
- D. Access the Table Configuration, add the field, and then modify the form layout

Answer: D

Explanation: The correct sequence involves accessing the Table Configuration to add the field and then modifying the form layout to include that new field.

Question: 1247

Washington DC scoped app "HROnboardFlow" uses Flow Designer with data pills

{{step.input.u_doc_type}}}. Fix for pill reference error adds fallback. Capture excluding subflow inputs. What pill filter?

- A. Branch pill.
- B. Set 'PillFix', pill refs.
- C. Manual.
- D. Filter: sys_flow_action ONLY, add || 'default', ^subflow.

Answer: D

Explanation: Washington DC's 'sys_flow_action ONLY' captures data pills {{step.input.u_doc_type || 'default'}}}, excluding subflows via ^ for reference integrity.

Question: 1248

An application developer needs to create a new table that requires several reference fields to existing tables. What is the best practice for managing these relationships?

- A. Create reference fields directly in the new table without any additional considerations.
- B. Use a single reference field to link to a parent table and avoid multiple references.
- C. Ensure that all reference fields are indexed for performance optimization.
- D. Define foreign key constraints in the database for data integrity.

Answer: C

Explanation: Indexing reference fields is a best practice as it enhances performance when querying related records, especially in applications with complex relationships.

Question: 1249

A query business rule on 'u_audit_log' must limit to logs created by user or in user's department within 90 days, for 'audit_viewer' role, using gs.endOfLastMonth() for date range, but skip if query count < 10 via GlideAggregate pre-check. What script?

- A.

```
var ga = new GlideAggregate('u_audit_log'); ga.addAggregate('COUNT'); ga.addEncodedQuery('created_by.department=' + gs.getUser().getDepartmentID() + '^ORcreated_by=' + gs.getUserID() + '^created_on>=90 days ago'); ga.query(); if (ga.next() && ga.getAggregate('COUNT') < 10) return; if (gs.getUser().hasRole('audit_viewer')) current.addEncodedQuery('created_by=' + gs.getUserID() + '^ORcreated_by.department=' + gs.getUser().getDepartmentID() + '^created_on>=' + gs.daysAgoStart(90));
```
- B.

```
if (!gs.getUser().hasRole('audit_viewer')) { current.addQuery('sys_created_on', '>=',
```

```

gs.daysAgoStart(90)).addQuery('created_by',
gs.getUserID()).addOrCondition('created_by.department',
gs.getUser().getDepartmentID()); }
C. (function() { var countGa = new GlideAggregate('u_audit_log');
countGa.setLimit(10); countGa.addQuery('sys_created_on', '>=', gs.daysAgoStart(90));
countGa.query(); if (countGa.getRowCount() >= 10) { if
(gs.getUser().hasRole('audit_viewer')) { current.addQuery('created_by.department',
gs.getUser().getDepartmentID()).addOrCondition('created_by',
gs.getUserID()).addQuery('sys_created_on', '>=', gs.daysAgoStart(90)); } } })();
D. var cutoff = gs.daysAgoStart(90); if (gs.getUser().hasRole('audit_viewer'))
current.addEncodedQuery('(created_by=' + gs.getUserID() +
'^ORcreated_by.department=' + gs.getUser().getDepartmentID()) ^sys_created_on>=' +
cutoff);

```

Answer: D

Explanation: The script in option C directly applies the encoded query with OR conditions and 90-day cutoff for audit viewers, efficient without pre-count which adds overhead in query business rules. Option A pre-queries unnecessarily, B misses OR properly, and D uses setLimit on aggregate incorrectly.

Question: 1250

To ensure that a business rule only runs when a specific field is modified, which condition should be used?

- A. current.field != previous.field;
- B. previous.field == current.field;
- C. current.field.changes();
- D. current.field.changes();

Answer: C

Explanation: The correct condition to ensure a business rule only runs when a specific field is modified is to use `current.field.changes()`, which checks if the field has changed.

Question: 1251

For 'u_it_asset_request' form, add 'u_justification' with char limit 1000, remove 'sys_created_on' from standard view, UI Policy sets 'u_approval_level' to 'manager' default if cost >1000. Client script onChange 'u_cost' currency: if value>1000 g_form.setValue('u_approval_level','manager'); GlideAjax 'ApprovalRouter.route' param

cost, set 'u_routed_to' reference to response user sys_id, showInfoMessage('Routed to ' + display name). Parse user name from XML.

- A. Text; remove policy; Policy simple; onLoad route
- B. String no limit; hide layout; condition >1000; script onChange setValue, Ajax attr
- C. Dictionary u_justification max_length=1000; standard view remove created_on; UI Policy condition u_cost >1000 default_value='manager' on u_approval_level; onChange u_cost: if (parseFloat(value)>1000) g_form.setValue('u_approval_level','manager'); var ga = new GlideAjax('ApprovalRouter'); ga.addParam('sysparm_name','route'); ga.addParam('sysparm_cost',value); ga.getXMLAnswer(function(r){ var userId = r.responseXML.documentElement.getAttribute('user_id'); var userName = r.getAttribute('user_name'); g_form.setValue('u_routed_to', userId); g_form.addInfoMessage('Routed to ' + userName); })
- D. Integer; view; condition cost.changesTo >1000; server

Answer: C

Explanation: Max_length enforces justification brevity. Standard view hides audit. Policy defaults level on cost. onChange triggers if, Ajax routes to user, sets reference and messages name for transparency.

Question: 1252

In a multi-domain ServiceNow environment, how can a developer ensure that users only see records relevant to their domain?

- A. Use global access controls for all records.
- B. Implement domain-specific access controls on tables.
- C. Create a single view for all users.
- D. Rely on user roles to filter records.

Answer: B

Explanation: Implementing domain-specific access controls on tables ensures that users only see records relevant to their domain, maintaining data privacy and security.

Question: 1253

When designing a data schema for a custom application, which of the following is a recommended practice for managing large datasets?

- A. Normalize the database to reduce redundancy.

- B. Use single-table inheritance to simplify the schema.
- C. Denormalize the database for faster read operations.
- D. Avoid using reference fields to prevent complex joins.

Answer: A

Explanation: Normalizing the database helps to reduce redundancy and improve data integrity, which is crucial for managing large datasets effectively.

Question: 1254

Debugging a denial on update for 'u_incident_extensions' table reveals that a business rule inserting extensions via GlideAjax from client script fails with "Security restricted: write denied". The client script: `var ga = new GlideAjax('MyUtils');`
`ga.addParam('sysparm_name', 'updateExtension');`
`ga.addParam('table', 'u_incident_extensions');`
`ga.getXMLAnswer();`. Server-side MyUtils uses `gs.getUser().hasRole('itil')` but ignores table ownership. To secure this API endpoint against unauthorized updates, which scripted fix incorporates role and ownership checks?

- A. Add to client: `g_user.hasRoleExactly('itil')` before GlideAjax, and server: `answer = gs.getUser().hasRole('itil') && current.incident.caller_id == gs.getUserID();`
- B. In MyUtils script include: `if (!gs.getUser().hasRole('itil')) return;`
`var gr = new GlideRecordSecure('u_incident_extensions');`
`gr.addQuery('incident', input.incident_sys_id);`
`gr.query();`
`if (gr.next() && gr.assigned_to == gs.getUserID()) {`
`gr.update();`
`}`
- C. Create write ACL: script: `answer = gs.getUser().hasRole('itil') && gs.getUser().getLocation() == current.location;`
- D. Use `gs.eventQueue()` in client to trigger server event with:
`payload.hasOwnProperty('incident') && gs.getUser().hasRole('itil');`

Answer: B

Explanation: GlideAjax callbacks must use GlideRecordSecure for ACL enforcement in server-side updates, ensuring itil role and ownership (assigned_to match) prevent unauthorized modifications. This secures the API by validating queries before updates, avoiding denials from loose role checks. Debugging via logs identifies the restriction; the fix tests for record existence and user assignment, embodying secure scripting best practices without over-permissive event queues.

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 syllabus 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.