NetSuite: SuiteCloud Developer II Sample Test

2018

Copyright © 2020

Contents

About this Sample Test 5

Subject Areas Covered by the Test: 6

I. Select the SuiteScript 2.0 code snippet that implements a described business process. 9

II. Determine which variables would be good candidates to use as a company preference, user preference, or Script Deployment Parameter. 10

III. Identify the impact of execution context on user event scripts 10

IV. Describe the ways in which conditions and triggers can be configured to execute actions and transitions. 11

V. Identify implications of search techniques in SuiteTalk and SuiteScript. 12

VI. Determine the use of SuiteCloud Processors 12

VII. Identify the functionality and best practices when using the Map/Reduce script type 12

VIII. Calculate the governance of a script 13

IX. Identify implications of deploying multiple User Event scripts against single record, order of execution, how user events interact with other records. 13

X. Recognize script deployment configuration across script types. 14

XI. Identify how to configure standard actions and define custom actions. 15

XII. Identify how to control the User Interface throughout the workflow lifecycle 15

XIII. Identify the functionality of record and workflow fields. 15

XIV. Compare capabilities of SuiteFlow and SuiteScript and determine when to use which technology. 15

XV. Identify the SuiteTalk (Web Services) support, versioning, deprecation and General Availability policy of endpoints. 16

XVI. Identify the impact of scripts and workflows on integrations. 16

XVII. Identify the process for upgrading SuiteTalk (Web Services) endpoints 17

XVIII. Determine session policy for Web Services vs UI, and how sessions are managed for the same user across multiple integrated applications. 17

XIX. Identify how to develop Data Center agnostic integrations. 17

XX. Determine the appropriate search technique to use in SuiteTalk (Web Services). 17

XXI. Identify how asynchronous and synchronous Web Services APIs impact integration design and implementation. 18

XXIII. Identify implications of managing a large volume of search results. 18

XXIV. Identify implications of search techniques in SuiteTalk and SuiteScript. 19

XXV. Identify strategies for and implications of role management and authentication when integrating with external systems. 19

XXVI. Identify NetSuite functionality and recommended practices related to restricted data fields (PII, PCI, e.g.) 20

XXVII. Determine how to optimize performance, scalability, and reliability in SuiteScript, SuiteFlow and SuiteTalk. 20

XXVIII. Determine how to trouble-shoot and debug in SuiteScript, SuiteFlow and SuiteTalk. 21

XXIX. Given a scenario, identify applicable integration technologies and their implications. 21

XXX. Determine how SuiteBundler handles collisions or conflicts during installation or update. 21

XXXI. Identify how to detect and prevent duplicate records. 21

XXXIII. Identify the impact of an account refresh on bundles 22

XXXV. Identify the performance implications of adding custom fields and the strategies to mitigate performance impact. 23

XXXVI. Given a scenario, select the sourcing and filtering criteria, or the defaulting and validation options for custom fields 24

XXXVII. Identify record-locking behavior and options 24

XXXVIII. Identify the various options available under the permissions model for custom records 24

XXXIX. Identify implications of deleting or inactivating custom record types. 24

XL. Recognize the purpose of Plug-Ins in SuiteScript (2.0). 25

XLI. Identify the functionality and capabilities of SuiteScript debugger 25

XLII. Identify the risks of implementing only client-side validations and strategies to address them. 25

XLIII. Identify how to dynamically scale a scheduled script utilization to match available queues in account. 25

XLIV. (new objective) Identify methods for outbound communication and data synchronization from NetSuite. 25

XLV. Determine how to interact with custom child record sublists in SuiteScript (2.0). 25

XLVI. Determine how to dynamically customize UI field attributes in SuiteScripts (2.0). 25

XLVII. Identify the timezone implications of various ways of setting date / time values and the ways in which these values are interpreted in SuiteScript (2.0) 26

XLVIII. Identify the capabilities of UI Objects. 26

XLIX. Identify capabilities and controls available with states and branching. 26

L. Identify the usage of scheduling in workflow. 26

LI. Determine how to use formulas in SuiteFlow 26

LII. Identify the role of installation scripts in SuiteBundler. 26

LIII. Identify implications of using saved searches and coded searches. 26

LIV. Determine how to set roles and permissions for a given situation in various environments. 26

LV. Identify considerations when working with production and non-production environments (e.g. differences in behaviour, testing, customizations, etc.) 26

LVI. Answer Keys 27

About this Sample Test

None of these questions are on the SuiteCloud Developer II Exam. Most of these questions were written by the same authors who wrote the actual exams. We have tried to write them to a similar level of difficulty as the actual exam questions, but do not guarantee this.

This sample test is provided to give you an idea of the format of questions that might be asked on the exam. However, the sample test does not provide comprehensive coverage of all topics on the test. All material listed in the SuiteCloud Developer II Study Guide may be tested.

Being able to answer all the below questions correctly does not guarantee that you will pass the SuiteCloud Developer II Exam exam.

We do not publish our passing score.

The number of sample questions per test objective does not reflect the actual weighting of the exam. Any objective may have more, or fewer, questions than are presented here. All objectives have at least one sample question included here.

Subject Areas Covered by the Test:

These are the broad subject areas covered on the Exam:

- SuiteScript (2.0)
- Analytics
- SuiteFlow
- SuiteTalk
- SuiteBundler
- SuiteBuilder
- Design Fundamentals

Below is each subject area broken out into more detail:

Topic	Objective
SuiteScript 2.0	**Select the SuiteScript (2.0) code snippet that implements a described business process.**
	Recognize the purpose of Plug-Ins in SuiteScript (2.0).
	Identify the functionality and capabilities of SuiteScript debugger
	Given a scenario, identify the proper pattern to invoke a scheduled script to support a long-running process.
	Calculate the governance of a script
	Identify implications of deploying multiple User Event scripts against a single record, order of execution, and how user events interact with other records in SuiteCloud technologies.
	Identify the risks of implementing only client-side validations and strategies to address them.
	Identify how to dynamically scale a scheduled script utilization to match available queues in account.
	Recognize script deployment configuration across script types.
	Determine which variables would be good candidates to use as a company preference, user preference, or Script Deployment Parameter.
	Identify the impact of execution context on user event scripts.
	(new objective) Identify methods for outbound communication and data synchronization from NetSuite.
	Determine how to interact with custom child record sublists in SuiteScript (2.0).
	Determine how to dynamically customize UI field attributes in SuiteScripts (2.0).
	Identify the timezone implications of various ways of setting date / time values and the ways in which these values are interpreted in SuiteScript (2.0)
	(new objective) Determine the use of SuiteCloud Processors
	(new objective) Identify the functionality and best practices when using the Map/Reduce script type
	Identify the capabilities of UI Objects.
SuiteFlow	**Identify how to configure standard actions and define custom actions.**

	Identify how to control the User Interface throughout the workflow lifecycle.
	Describe the ways in which conditions and triggers can be configured to execute actions and transitions.
	Identify capabilities and controls available with states and branching.
	Identify the functionality of record and workflow fields.
	Compare capabilities of SuiteFlow and SuiteScript (2.0) and determine when to use which technology.
	Identify the usage of scheduling in workflow.
	Determine how to use formulas in SuiteFlow.
SuiteTalk	**Identify the SuiteTalk (Web Services) support, versioning, deprecation and General Availability policy of endpoints.**
	Identify the process for upgrading SuiteTalk (Web Services) endpoints.
	Identify SuiteTalk (Web Services) authentication methods and benefits.
	Determine session policy for Web Services vs UI; and how sessions are managed for the same user across multiple integrated applications.
	Determine session and thread management techniques with a SuiteCloud Plus license.
	Identify how to develop Data Center-agnostic integrations.
	Determine the appropriate search technique to use in SuiteTalk (Web Services).
	Identify how asynchronous and synchronous Web Services APIs impact integration design and implementation.
	Identify how to build efficient data synchronization.
SuiteBuilder	**Determine the impact of form customization on records.**
	Identify the performance implications of adding custom fields and the strategies to mitigate performance impact.
	Given a scenario, select the sourcing and filtering criteria, or the defaulting and validation options for custom fields.
	Identify record-locking behavior and options.
	Identify the various options available under the permissions model for custom records.
	Identify how configurations, settings, and preferences impact the behavior of a NetSuite instance.
	Identify implications of deleting or inactivating custom record types.
SuiteAnalytics	**Identify implications of using saved searches and coded searches.**
	Identify implications of managing a large volume of search results.
	Identify implications of search techniques in SuiteTalk and SuiteScript (2.0).
Design Fundamentals	**Identify the role of installation scripts in SuiteBundler**.
	Determine how SuiteBundler handles collisions or conflicts during installation or update.

	Identify the impact of an account refresh on bundles.
	Determine how to set roles and permissions for a given situation in various environments.
	Identify strategies for and implications of role management and authentication when integrating with external systems.
	Identify NetSuite functionality and recommended practices related to restricted data fields (PII, PCI, e.g.).
	Determine how to optimize performance, scalability, and reliability in SuiteScript (2.0), SuiteFlow and SuiteTalk.
	Determine how to trouble-shoot and debug in SuiteScript (2.0), SuiteFlow and SuiteTalk.
	Identify considerations when working with production and non-production environments (e.g. differences in behaviour, testing, customizations, etc.)
	Given a scenario, identify applicable integration technologies and their implications.
	Identify how to detect and prevent duplicate records.

I. Select the SuiteScript 2.0 code snippet that implements a described business process.

1. A "Media Value" custom field is needed to show the customer the retail price of a service while the standard NetSuite fields show their discounted price. The amount for this field will be the item base price multiplied by the quantity.

   Which code will address this process?

   A.

```
function validateLine(scriptContext) {
      if (scritpContext.sublistId == 'item') {
         var salesOrder = scriptContext.currentRecord;
         var basePrice = // insert a valid process to get the base
price
         var qty = salesOrder.getCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'quantity',
         });

         salesOrder.setCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'custcol_media_value',
            value    : qty * basePrice
         });
      }
      return true;
}
```

   B.

```
function validateLine(scritpContext) {
      if (scritpContext.sublistId == 'item') {
         var salesOrder = scritpContext.currentRecord;
         var basePrice = salesOrder.getCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'rate',
         });
         var qty = salesOrder.getCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'quantity',
         });

         salesOrder.setCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'custcol_media_value',
            value    : qty * basePrice
         });
      }
      return true;
}
```

Copyright © 2020

C.

```
function validateLine(scritpContext) {
      if (scritpContext.sublistId == 'item') {
         var salesOrder  = scritpContext.currentRecord;
         var basePrice = salesOrder.getCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'amount',
         });
         var qty = salesOrder.getCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'quantity',
         });

         salesOrder.setCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'custcol_media_value',
            value    : qty * basePrice
         });
      }
      return true;
}
```

D.

```
function validateLine(scritpContext) {
      if (scritpContext.sublistId == 'item') {
         var salesOrder = scritpContext.currentRecord;
         var basePrice = // insert a valid process to get the base
price
         var qty = salesOrder.getCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'quantity',
         });

         salesOrder.setCurrentSublistValue({
            sublistId: 'item',
            fieldId  : 'rate',
            value    : qty * basePrice
         });
      }
      return true;
}
```

II. Determine which variables would be good candidates to use as a company preference, user preference, or Script Deployment Parameter.

2. Assuming that you have multiple script deployments for a given scheduled script definition with script parameters, identify capabilities of using script parameters.

   A. Provides the ability to change script behavior based on the script deployment
   B. Allows passing of NetSuite objects without the use of JSON.stringify()
   C. Allows override of parameter values when using N/task module.
   D. Allows definition of audience based on script deployment

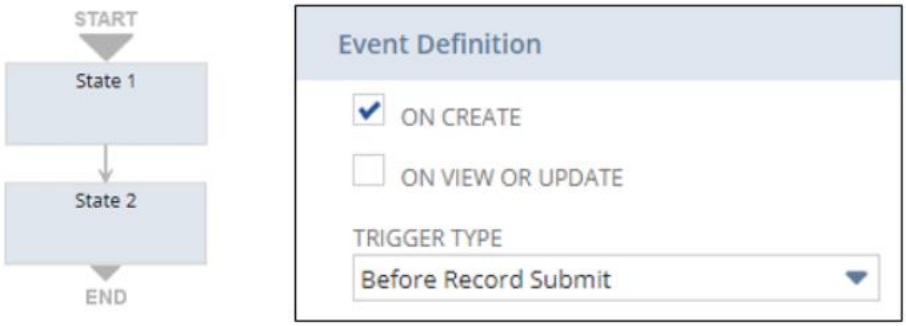
III. Identify the impact of execution context on user event scripts

3. What is an expected behavior for user event scripts designed to execute on the "WEBSTORE" context?

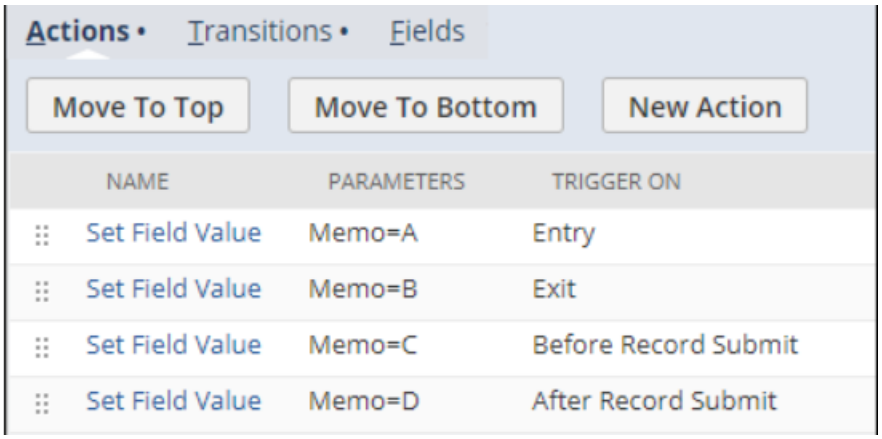
   A. CSV Import of web orders will execute scripts on the sales order record.
   B. SuiteTalk integration for web orders will execute scripts on the sales order record.
   C. Scripts on the sales order will execute for orders submitted through the web store only.
   D. A sales rep creating a sales order thru the user interface will trigger the scripts if the order is for a web customer.

IV. Describe the ways in which conditions and triggers can be configured to execute actions and transitions.

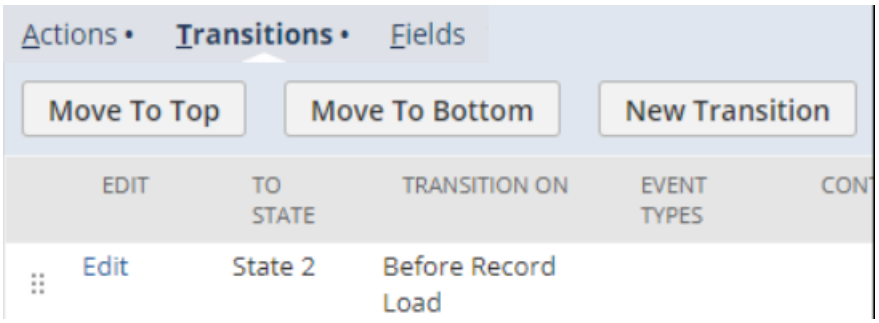
4. Evaluate this screenshot of workflow initiation, states, actions, and transitions:



Actions on State 1. There are no actions in State 2.



Transition:



What is the final value of Memo after the end user clicks Save on a new record?

A. A
B. B
C. C
D. D

V. Identify implications of search techniques in SuiteTalk and SuiteScript.

5. What is the most reliable way to find all records with a given email address?

   A. search.global() with the email address as the keyword
   B. search.create() on the entity record using the email filter
   C. search.load() on the entity record using the email filter
   D. search.global() on customer, contact, and employee records using the email filter

VI. Determine the use of SuiteCloud Processors

6. Which statement is true about SuiteCloud Processors?

   A. A SuiteCloud Processor can only be used with SuiteScript 2.0 scripts.
   B. SuiteCloud Processors replace the use of script queues in scheduled jobs, like those of scheduled scripts.
   C. The scheduler for SuiteCloud Processors provide load-balancing by automatically determining the order in which jobs start to process.
   D. SuiteCloud Processors is the next iteration of the SuiteCloud Plus upgrade making the upgrade unnecessary.
   E. SuiteCloud Processors increase the performance of an individual unit of work.

VII. Identify the functionality and best practices when using the Map/Reduce script type

7. How many soft limit usage units does each map/reduce job have?

A. 1,000
B. 5,000
C. 10,000
D. 20,000
E. No governance limits

VIII. Calculate the governance of a script

8. Given the following scheduled script (10,000 usage unit limit):

```
function updateSalesOrders() {
    var soSearch = search.load({id : 'customsearch_orders_to_update'});
    var results = soSearch.run().getRange({
       start : 0,
       end   : 1000
    });
    record.submitFields({
       type : record.Type.SALES_ORDER,
       id   : results[0].getValue({name : 'internalid'}),
       values : {custbody_approved : true}
    });
    var script = runtime.getCurrentScript();
    var usageRemaining = script.getRemainingUsage();
 }
```

API usage unit:

- seach.load() - 5 units
- Search.run().getRange() - 10 units
- record.submitFields() - 10 units on transactions; 5 units on non-transactions

What is the value of "usageRemaining" after a single execution of the script?

A. 9,955
B. 9,975
C. 9,980
D. 10,000

IX. Identify implications of deploying multiple User Event scripts against single record, order of execution, how user events interact with other records.

9. Assuming the scripts below are the only user event scripts deployed to the 'Employee' and 'customrecord_reward' records.

   Script #1. User Event Script with After Submit function deployed to the Employee record

```
function afterSubmit(scritpContext) {
      var reward = record.create({
         type      : 'customrecord_reward',
         isDynamic : true
      });
```

```
    reward.setValue({
       fieldId : 'custrecord_myrecordfield1',
       value   : 'my value1'
    });
    var recordId = reward.save({ignoreMandatoryFields : true});
}
```

Script #2. User Event Script with Before Load, Before Submit and After Submit functions deployed to the 'customrecord_reward' record

```
function beforeLoad() {
    log.debug({title : 'Before Load Function'});
}
function beforeSubmit() {
    log.debug({title : 'Before Submit Function'});
}
function afterSubmit() {
    log.debug({title : 'After Submit Function'});
}
```

When the After Submit function on the employee record is triggered, how many times will the log.debug() statement be executed?

A. 0
B. 1
C. 2
D. 3

X. Recognize script deployment configuration across script types.

10. The script below has been deployed as a Save Record function at the form level.

```
1  function saveRecord(context){
2     var custObj = context.currentRecord;
3     var companyId = custObj.getValue({fieldId: 'entity'});
4     var suiteletUrl = url.resolveScript({
5        scriptId    : 'customscript1',
6        deploymentId: 'customdeploy1',
7        returnExternalUrl: false
8     });
9     custobj.setValue({
10       fieldId : 'custbody10',
11       value   : urlSuitelet + '&custpage_companyid=' + companyId
12    });
13    log.debug({title: 'Request to the Suitelet has been sent'});
14    return true;
15 }
16
```

Which line will not work?

A. Line 3
B. Line 4
C. Line 9
D. Line 13

XI. Identify how to configure standard actions and define custom actions.

11. Which action can be used to create an estimate transaction from a customer workflow?

    A. Use the Create Record action and select estimate.
    B. Use a custom action using SuiteScript to create an estimate.
    C. Use the Go To Record action and go to a new estimate record.
    D. Use the Transform Record action and transform to an estimate.

XII. Identify how to control the User Interface throughout the workflow lifecycle

12. A particular workflow has three states:

    State A > State B > State C > (exit workflow)

    The Lock Record action is triggered in State B.

    Which statement describes how this action will affect the record?

    A. The record will remain locked while in State B, and will be unlocked upon entering State C.
    B. The record will remain locked while in State B and State C, and will stay locked after it exits the workflow.
    C. The record will remain locked while in State B and State C, and will be unlocked when the record exits the workflow.

XIII. Identify the functionality of record and workflow fields.

13. What type of field can hold data and is accessible throughout a workflow, regardless of the state?

    A. create a state custom field and check store value
    B. create a state custom field and uncheck store value
    C. create a workflow custom field and select store value
    D. create a workflow custom field and unselect store value

XIV. Compare capabilities of SuiteFlow and SuiteScript and determine when to use which technology.

14. An end user clicks Save to create a sales order. The system displays a new sales order form with default values for Customer, Memo, and Location from the sales order just saved.

Which option allows this automation to be implemented without requiring additional actions or functions?

A. SuiteFlow Go To Page action
B. SuiteFlow Go To Record action
C. SuiteScript http.get() function
D. SuiteScript url.resolveRecord() function
E. SuiteScript redirect.toRecord() function

XV. Identify the SuiteTalk (Web Services) support, versioning, deprecation and General Availability policy of endpoints.

15. What process must be followed to upgrade to the newest WSDL version?

A. Jump to, test and upgrade to the desired version
B. Sequentially test and upgrade all WSDL versions between current one being used and newest version
C. Sequentially test and upgrade the "_1" (i.e. v2013_1_x) versions between current one being used and newest version
D. Sequentially test and upgrade the "_2" (i.e. v2013_2_x) versions between current one being used and newest version

XVI. Identify the impact of scripts and workflows on integrations.

16. An e-commerce (SuiteTalk) integration is not properly submitting new Sales Orders to a NetSuite account. Each add request of Sales Order is returning this response:

```
<soapenv:Body>
 <addResponse xmlns="urn:messages_2014_1.platform.webservices.netsuite.com">
        <writeResponse>
              <platformCore:status isSuccess="false" xmlns:platformCore="urn:core_2014_1.platform.webservices.netsuite.com">
                <platformCore:statusDetail type="ERROR">
                      <platformCore:code>USER_ERROR</platformCore:code>
                      <platformCore:message>Your search request has timed out. You may need to refine your search or combine the results of
multiple searches to obtain the desired result.</platformCore:message>
                </platformCore:statusDetail>
              </platformCore:status>
        </writeResponse>
 </addResponse>
</soapenv:Body>
```

What should be done to prevent the error?

A. Lower the search Page Size on the Web Services Preferences.
B. Set the bodyFieldsOnly search preference to true on each request.
C. Fix the search that is timing out on the user event script deployed to Sales Order.

D. Change the search that is being used to initiate a scheduled workflow on Sales Order.
E. Set the application time out to a number higher than 180000 milliseconds (180 seconds).

XVII. Identify the process for upgrading SuiteTalk (Web Services) endpoints

17. Which three conditions will require parts of an integration to be re-coded when the NetSuite WSDL is updated to the newest version?

A. The data type for a field has changed
B. A fix to a defect was released in the latest version
C. A field that is referenced in an integration has been removed from NetSuite
D. A field that was required in the previous version is optional in the latest version
E. A field that was optional in previous versions of NetSuite becomes required in the latest version

XVIII. Determine session policy for Web Services vs UI, and how sessions are managed for the same user across multiple integrated applications.

18. What happens to an Active UI Session when the same user is logged in using WebService Integrated application?

A. The Web Service Integrated application logs in simultaneously.
B. It sends notification to the UI session for another application login.
C. It kicks the UI session out because only one session is allowed per user.
D. NetSuite Security Algorithm locks down the account to avoid a possible user hack.

XIX. Identify how to develop Data Center agnostic integrations.

19. What is the recommended procedure, using the latest web services endpoint, to successfully connect to any NetSuite account regardless of the data center location?

A. Include all NetSuite web services domains in SOAP requests.
B. Go to web services preferences page to retrieve the relevant domain.
C. Create a RESTlet that will return the data center where the account is located.
D. Include a getDataCenterUrls request in the integration code to dynamically discover the correct URL for web services access for each account.

XX. Determine the appropriate search technique to use in SuiteTalk (Web Services).

20. Which API in SuiteTalk can be used to programmatically look for the correct saved search reference?

    A. getAll()
    B. getList()
    C. getSavedSearch()
    D. getCustomizationId()

XXI. Identify how asynchronous and synchronous Web Services APIs impact integration design and implementation.

21. Which three customization objects are supported by getCustomizationId?

    A. Custom forms
    B. Custom item fields
    C. Transaction item options
    D. Transaction column fields

XXII. Identify how to build efficient data synchronization.

22. Employee data from an external HRMS needs to be synchronized into NetSuite.

    Which technique will result in the least amount of SuiteTalk API calls?

    A. Store the NetSuite employee records internal IDs in the external HRMS employee table.
    B. Store the external HRMS employee table's primary keys in the NetSuite employee record's external ID field.
    C. Store the external HRMS employee table's primary keys in a custom field on the NetSuite employee record.
    D. Build a custom record type as a mapping table between NetSuite employee records and the external HRMS employee table.

XXIII. Identify implications of managing a large volume of search results.

23. Given the following initial response for a search request:

```
<soapenv:Envelope>
  <soapenv:Header>
    <platformMsgs:documentInfo>
    <platformMsgs:nsId>WEBSERVICES_528736_07012008543995006307049233_d53ef4d2273b15</platformMsgs:nsId>
    </platformMsgs:documentInfo>
  </soapenv:Header>
  <soapenv:Body>
    <platformMsgs:searchResponse>
     <platformCore:searchResult>
     <platformCore:status isSuccess="true"/>
     <platformCore:totalRecords>678</platformCore:totalRecords>
     <platformCore:pageSize>500</platformCore:pageSize>
     <platformCore:totalPages>2</platformCore:totalPages>
     <platformCore:pageIndex>1</platformCore:pageIndex>
     <platformCore:searchId>WEBSERVICES_528736_07012008543995006307049233_d53ef4d2273b15
              </platformCore:searchId>
     <platformCore:recordList>
      .....
     </platformCore:recordList>
     </platformCore:searchResult>
    </searchMoreWithIdResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

How many records will be returned by a subsequent searchMore operation if 25 records were deleted and 20 records were added prior to the searchMore request being submitted?

A. 153 records as deleted records will not be returned and new records will be ignored.
B. 173 records as the data returned is current such that if a change occurs in between the time of the original search operation and the searchMore operation, the updated data is returned.
C. 178 records as the results returned in a searchMore operation reflect the records in the next segment as defined in the original search operation.
D. 198 records as new records will be included in the response.

XXIV. Identify implications of search techniques in SuiteTalk and SuiteScript.

24. What is the most efficient and reliable approach to implement ad-hoc logical OR operator between search filters during the script execution time using SuiteScript?

A. The OR logical operator is not supported by SuiteScript.
B. Use nlapiLoadSearch().setFilterxxx) followed by nlobjSearch.runSearch().
C. Use nlapiCreateSearch().setFilterxxx) followed by nlobjSearch.runSearch().
D. Use nlapiSearchRecord() calls to get all the data, put it in Array objects and filter them in memory.

XXV. Identify strategies for and implications of role management and authentication when integrating with external systems.

25. The preferred method of building integrations that require the NetSuite application to access external systems is to use SuiteSignOn (outbound Single Sign-on).

    What is an acceptable alternative for storing passwords for external systems in NetSuite?

    A. Use SAML Single Sign-on
    B. Use the addCredentialField() SuiteScript API
    C. Use the Form.addField() API with type = "password" together with the N/crypto module
    D. Use the Form.addField() API with type = ''password" but only store a hash of the password and not the password itself

XXVI. Identify NetSuite functionality and recommended practices related to restricted data fields (PII, PCI, e.g.)

26. Which standard data field is displayed as masked by default for the Administrator role?

    A. Password
    B. Credit Card Number
    C. Social Security Number
    D. Driver's License Number

XXVII. Determine how to optimize performance, scalability, and reliability in SuiteScript, SuiteFlow and SuiteTalk.

27. Which scenario warrants the use of asynchronous web service operations?

    A. The connection to NetSuite is expected to be slow or unstable
    B. Processing transactions along with its related record
    C. When the returning a large number of results
    D. Long-running jobs, especially those exceeding the synchronous timeout limit of 15 minutes

28. What is the advantage of SuiteCloud Plus for SuiteTalk throughput capacity?

    A. It speeds up integration response time.
    B. It removes optimization algorithms for GUI requests.
    C. It leverages dedicated hardware for processing integration requests.
    D. It supports concurrent web services threads from a single named user.

Copyright © 2020

XXVIII. Determine how to trouble-shoot and debug in SuiteScript, SuiteFlow and SuiteTalk.

29. Which is a requirement to debug an existing script in the NetSuite Script Debugger?

    A. Set Log Level to Debug
    B. Set script deployment status to Testing
    C. Obtain a sandbox account
    D. Insert "debugger" statement in the script

XXIX. Given a scenario, identify applicable integration technologies and their implications.

30. Which one of the following are key strengths that SuiteTalk web services offer as an integration interface?

    A. Strictly defined data schema allows developers to focus on writing business logic
    B. Native JSON support allows quick and easy integration development
    C. Support for multiple secure authentication methods
    D. Multi-threading support that broadens integrated application's I/O bandwidth
    E. Can be easily deployed using SuiteBundler
    F. Complete freedom to define the supported operations

XXX. Determine how SuiteBundler handles collisions or conflicts during installation or update.

31. How can object conflicts and object overrides be avoided?

    A. Update Source account and lock all scripts into target account
    B. Update Target account and lock all scripts into Source account
    C. Update Target account object and let the bundle owner know what was modified
    D. Update the object in Source and process the bundle update on the target account

XXXI. Identify how to detect and prevent duplicate records.

32. Which statements are true for NetSuite's duplicate detection feature?

    A. Duplicates can be identified on the Lists > Mass Update > Entity Duplicate Resolution page.
    B. Duplicates can be identified on a subtab of the Setup > Company > Duplicate Detection page.
    C. Duplicates are tagged when all fields configured in the Set Up Duplicate Detection page match.

Copyright © 2020

D. Duplicates are tagged when any one field configured in the Set Up Duplicate Detection page match.

XXXII. Identify how configurations, settings, and preferences impact the behavior of a NetSuite instance.

33. Consider the code snippet below.

```
var isMultiCurrencyOn = runtime.isFeatureInEffect({feature : 'MULTICURRENCY'});
try {
   log.debug({title : 'START HERE'});
   if (isMultiCurrencyOn) {
      log.debug({title : 'WITH MULTICURRENCY'});
   }
   var tranSrch = search.create({
      type    : search.Type.TRANSACTION,
      columns : [ search.createColumn({name : 'mainline'}),
                  search.createColumn({name : 'debitamount'}),
                  search.createColumn({name : 'creditamount'}),
                  search.createColumn({name : 'fxamount'})
                ]
   });
   var results = tranSrch.run().getRange({
      start : 0,
      end   : 1000
   });
   log.debug({title: 'END HERE'});
} catch (e) {
   log.debug({title: 'THERE WAS AN ERROR CAUGHT'});
}
```

If the above code is run in the Script Debugger of a Netsuite account, and the value of 'isMultiCurrencyOn' variable is false, what log.debug() lines are called?

A. log.debug({title: 'START HERE'});
B. log.debug({title: 'END HERE'});
C. log.debug({title: 'THERE WAS AN ERROR CAUGHT'});
D. log.debug({title: 'WITH MULTI CURRENCY'});

XXXIII. Identify the impact of an account refresh on bundles

34. Which two are recommended processes when using a sandbox account for bundle development and/or testing?

A. Create the bundle in sandbox, then install it in production.
B. Create the bundle in production, then refresh the sandbox.
C. Make further changes to the bundle in production. Refresh the sandbox for testing purposes.

Copyright © 2020

D. Make further changes to the bundle in sandbox. Manually replicate the changes in production.

XXXIV. Determine the impact of form customization on records.

35. A dedicated web services user has a dedicated custom role. An integration needs to be built to edit a custom entity field on the Customer record. No one other than the dedicated web services user should be able to view or manipulate that custom field.

    After ensuring the custom field is visible and editable through the preferred entry form of the web services user, which two additional steps are required to meet this objective?

    A. Set the default access level on the custom entity field to Edit.
    B. Set the default access level on the custom entity field to None.
    C. Set the web services dedicated custom role permission for Custom Fields to Edit.
    D. Set the access level on the custom entity field to edit for the dedicated web services role.
    E. Set the custom field as hidden on all UI forms, except the preferred entry form for the web services user.

XXXV. Identify the performance implications of adding custom fields and the strategies to mitigate performance impact.

36. On the contact record, the entityid field and email field are configured to be mandatory. There are many custom fields configured for the contact record. Most of these custom fields are linked via sourcing and have validation set. They are only useful for the end users, and do not hold sensitive data.

    An integration application imports contact records into NetSuite using SuiteTalk. This integration uses a tailored contact custom form which exposes the entityid field and email field, and hides all the custom fields.

    Which best describes the highest NetSuite server performance gain for this integration when using this custom form?

    A. The integration application can import contact records with the least possible amount of privileges.
    B. Without all the custom fields set, incoming SOAP requests are smaller and require less server resource to parse.
    C. Since no custom fields are set, the integration application does not need to make SuiteTalk customization API calls.
    D. Without all the custom fields set, the NetSuite server does not need to process their validation, sourcing and filtering.

XXXVI. Given a scenario, select the sourcing and filtering criteria, or the defaulting and validation options for custom fields

37. There are two custom List/Record fields on a record where the values of Field 2 are dependent on the selection in Field 1.

    How can this be accomplished?

    A. Define a Dynamic Default for Field 2.
    B. Define a Default Selection for Field 2.
    C. Use Field 1 as a Source List and choose a common attribute under Source Filter By.
    D. Define a filter and choose a common attribute for Filter Using, choose 'equal' as Compare Type and define Field 1 as the Compare To Field.

XXXVII. Identify record-locking behavior and options

38. When does the system check for conflicting updates whenever a user or script attempts to save updates to a NetSuite custom record instance?

    A. Always. NetSuite automatically provides optimistic locking for all custom record types.
    B. When the Enable Optimistic Locking option is enabled for the custom record type.
    C. When a user event script has been created that executes this check when custom records are saved.
    D. Record locking is not supported for custom record types.

XXXVIII. Identify the various options available under the permissions model for custom records

39. Which three access levels can be defined for custom fields?

    A. Roles
    B. Users
    C. Classes
    D. Locations
    E. Department
    F. Subsidiaries

XXXIX. Identify implications of deleting or inactivating custom record types.

40. For the custom record above, the Color field is a List Record field that selects from the Color custom record type.

Can the Red custom record value be deleted from the list of Color custom records?

A. No, custom record values can only be inactivated once they are referenced.
B. Yes, if all records that have Red referenced on them are first inactivated.
C. Yes, if the Color field has the Allow Delete of List/Record Values? property set to Prevent and Return Error.
D. Yes, if the Color field has the Allow Delete of List/Record Values? property set to Allow and Set Dependent Field Values to Null.

XL. Recognize the purpose of Plug-Ins in SuiteScript (2.0).

Sample Q not yet available. To be released by 11/30/2018

XLI. Identify the functionality and capabilities of SuiteScript debugger

Sample Q not yet available. To be released by 11/30/2018

XLII. Identify the risks of implementing only client-side validations and strategies to address them.

Sample Q not yet available. To be released by 11/30/2018

XLIII. Identify how to dynamically scale a scheduled script utilization to match available queues in account.

Sample Q not yet available. To be released by 11/30/2018

XLIV. (new objective) Identify methods for outbound communication and data synchronization from NetSuite.

Sample Q not yet available. To be released by 11/30/2018

XLV. Determine how to interact with custom child record sublists in SuiteScript (2.0).

Sample Q not yet available. To be released by 11/30/2018

XLVI. Determine how to dynamically customize UI field attributes in SuiteScripts (2.0).

Sample Q not yet available. To be released by 11/30/2018

Copyright © 2020

XLVII. Identify the timezone implications of various ways of setting date / time values and the ways in which these values are interpreted in SuiteScript (2.0)

Sample Q not yet available. To be released by 11/30/2018

XLVIII. Identify the capabilities of UI Objects.

Sample Q not yet available. To be released by 11/30/2018

XLIX. Identify capabilities and controls available with states and branching.

Sample Q not yet available. To be released by 11/30/2018

L. Identify the usage of scheduling in workflow.

Sample Q not yet available. To be released by 11/30/2018

LI. Determine how to use formulas in SuiteFlow

Sample Q not yet available. To be released by 11/30/2018

LII. Identify the role of installation scripts in SuiteBundler.

Sample Q not yet available. To be released by 11/30/2018

LIII. Identify implications of using saved searches and coded searches.

Sample Q not yet available. To be released by 11/30/2018

LIV. Determine how to set roles and permissions for a given situation in various environments.

Sample Q not yet available. To be released by 11/30/2018

LV. Identify considerations when working with production and non-production environments (e.g. differences in behaviour, testing, customizations, etc.)

Sample Q not yet available. To be released by 11/30/2018

LVI. Answer Keys

1	A
2	A and C
3	C
4	D
5	A
6	B and C
7	C
8	B
9	A
10	D
11	B
12	A
13	C
14	B
15	A
16	C
17	A, C and E
18	A
19	D
20	C

21	B, C and D
22	B
23	A
24	C
25	B
26	B
27	A and D
28	D
29	B
30	A
31	D
32	A and C
33	A and C
34	A and D
35	B and D
36	D
37	D
38	B
39	A, E and F
40	D