Splunk SPLK-1004 Practice Questions

Shared by Leach on 18-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A report named "Linux logins" populates a summary index with the search string sourcetype=linux_secure | sitop src_ip user. Which of the following correctly searches against the summary index for this data?

Options:

A- index=summary sourcetype='linux_secure' | top src_ip user
B- index=summary search_name='Linux logins' | top src_ip user
C- index=summary search_name='Linux logins' | stats count by src_ip user
D- index=summary sourcetype='linux_secure' | stats count by src_ip user

Answer:

C

Explanation:

The correct way to search against the summary index for this data is:

index=summary search_name='Linux logins' | stats count by src_ip user

Here's why this works:

Summary Index : Summary indexes store pre-aggregated data generated by scheduled reports or saved searches. To query this data, you must specify the index=summary and filter by the search_name field, which identifies the specific report that populated the summary index.

Aggregation : The original search used sitop, which is designed for summary indexing. When querying the summary index, you should use stats to aggregate the pre-aggregated data further.

Example:

index=summary search_name='Linux logins'

| stats count by src_ip user

Splunk Documentation on Summary Indexing:
https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Usesummaryindexing

Splunk Documentation on sitop:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/sitop

Question 2

Question Type: MultipleChoice

Where does the output of an append command appear in the search results?

Options:

A- Added as a column to the right of the search results.
B- Added as a column to the left of the search results.
C- Added to the beginning of the search results.
D- Added to the end of the search results.

Answer:

D

Explanation:

The output of the append command is added to the end of the current search results. This is useful for concatenating additional data from a subsearch.

Question 3

Question Type: MultipleChoice

If a nested macro expands to a search string that begins with a generating command, what additional syntax is needed?

Options:

A- Double tick marks around the nested macro.
B- A comma before the nested macro.
C- Square brackets around the nested macro.
D- A pipe character before the nested macro.

Answer:

C

Explanation:

When a nested macro expands to a search string that begins with a generating command, square brackets are required to ensure proper interpretation. Square brackets allow the nested macro to be treated as a subsearch or command.

Question 4

Question Type: MultipleChoice

How is a cascading input used?

Options:

A- As part of a dashboard, but not in a form.
B- Without notation in the underlying XML.
C- As a way to filter other input selections.
D- As a default way to delete a user role.

Answer:

C

Explanation:

A cascading input is used to filter other input selections in a dashboard or form, allowing for a dynamic user interface where one input influences the options available in another input.

Cascading Inputs:

Definition: Cascading inputs are interconnected input controls in a dashboard where the selection in one input filters the options available in another. This creates a hierarchical selection process, enhancing user experience by presenting relevant choices based on prior selections.

Implementation:

Define Input Controls:

Create multiple input controls (e.g., dropdowns) in the dashboard.

Set Token Dependencies:

Configure each input to set a token upon selection.

Subsequent inputs use these tokens to filter their available options.

Example:

Consider a dashboard analyzing sales data:

Input 1: Country Selection

Dropdown listing countries.

Sets a token $country$ upon selection.

Input 2: City Selection

Dropdown listing cities.

Uses the $country$ token to display only cities within the selected country.

XML Configuration:

<input type='dropdown' token='country'>

<label>Select Country</label>

<choice value='USA'>USA</choice>

<choice value='Canada'>Canada</choice>

</input>

<input type='dropdown' token='city'>

<label>Select City</label>

<search>

<query>index=sales_data country=$country$ | stats count by city</query>

</search>

</input>

In this setup:

Selecting a country sets the $country$ token.

The city dropdown's search uses this token to display cities relevant to the selected country.

Benefits:

Improved User Experience: Users are guided through a logical selection process, reducing the chance of invalid or irrelevant selections.

Data Relevance: Ensures that dashboard panels and visualizations reflect data pertinent to the user's selections.

Other Options Analysis:

B . As part of a dashboard, but not in a form:

Cascading inputs are typically used within forms in dashboards to collect user input. This option is incorrect as it suggests a limitation that doesn't exist.

C . Without token notation in the underlying XML:

Cascading inputs rely on tokens to pass values between inputs. Therefore, token notation is essential in the XML configuration.

D . As a default way to delete a user role:

This is unrelated to the concept of cascading inputs.

Conclusion:

Cascading inputs are used in dashboards to create a dependent relationship between input controls, allowing selections in one input to filter the options available in another, thereby enhancing data relevance and user experience.

Question 5

Question Type: MultipleChoice

How can the erex and rex commands be used in conjunction to extract fields?

Options:

A- The regex generated by the erex command can be edited and used with the rex command in a subsequent search.
B- The regex generated by the rex command can be edited and used with the erex command in a subsequent search.
C- The regex generated by the erex command can be edited and used with the erex command in a subsequent search.
D- The erex and rex commands cannot be used in conjunction under any circumstances.

Answer:

A

Explanation:

The erex command in Splunk generates regular expressions based on example data. These generated regular expressions can then be edited and utilized with the rex command in subsequent searches.

Question 6

Question Type: MultipleChoice

Which of the following will best optimize dashboard performance?

Options:

A- Use inline searches.
B- Use base searches.
C- Use accelerated data models.
D- Use scheduled reports.

Answer:

C

Explanation:

Accelerated data models in Splunk create summaries of data that can be queried more efficiently, significantly improving dashboard performance. By precomputing and storing results, dashboards can retrieve data faster, reducing load times and resource consumption.

According to Splunk Documentation:

'Data model acceleration speeds up reporting for the entire set of fields that you define in a data model and which you and your Pivot users want to report on.'

Question 7

Question Type: MultipleChoice

How can a lookup be referenced in an alert?

Options:

A- Use the lookup dropdown in the alert configuration window.
B- Follow a lookup with an alert command in the search bar.
C- Run a search that uses a lookup and save as an alert.
D- Upload a lookup file directly to the alert.

Answer:

C

Explanation:

In Splunk, a lookup can be referenced in an alert by running a search that incorporates the lookup and saving that search as an alert. This allows the alert to use the lookup data as part of its logic.

Question 8

Question Type: MultipleChoice

What is the value of base lispy in the Search Job Inspector for the search index=web clientip=76.169.7.252?

Options:

A- [index::web AND 169 252 7 76]
B- [AND 169 252 7 76 index::web]
C- [169 AND 252 AND 7 AND 76 index::web]
D- [index::web 169 AND 252 AND 7 AND 76]

Answer:

A

Explanation:

Comprehensive and Detailed Step by Step

The base lispy value in the Search Job Inspector represents the internal representation of the search query after it has been parsed and optimized by Splunk. It shows how Splunk interprets the query in terms of logical operations and field-value pairs.

For the search:

Copy

1

index=web clientip=76.169.7.252

The base lispy value will be:

Copy

1

[index::web AND 169 252 7 76]

Here's why this is correct:

Index Matching : The index::web part specifies that the search is scoped to the web index.

Field-Value Matching : The clientip field is broken down into its individual components (76, 169, 7, 252) for efficient matching using bloom filters and other optimizations.

Logical AND : Splunk combines these components with an AND operator to ensure all conditions are met.

Other options explained:

Option B : Incorrect because the order of AND and the components is incorrect.

Option C : Incorrect because the components are not properly grouped with the index.

Option D : Incorrect because the AND operator is misplaced, and the structure does not match Splunk's internal representation.

Splunk Documentation on Search Job Inspector:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Viewsearchjobproperties

Splunk Documentation on Bloom Filters:
https://docs.splunk.com/Documentation/Splunk/latest/Indexer/Bloomfilters

Question 9

Question Type: MultipleChoice

Which of the following attributes only applies to the form element, and not the dashboard root element of a SimpleXML dashboard?

Options:

A- hideEdit
B- hideTitle
C- hideFilters
D- hideChrome

Answer:

C

Explanation:

In Splunk's Simple XML, certain attributes are specific to the <form> element and do not apply to the <dashboard> root element. The hideFilters attribute is one such attribute that is exclusive to the <form> element. It controls the visibility of form input elements (filters) in the dashboard.

Setting hideFilters='true' within the <form> element hides the input fields, allowing for a cleaner dashboard view when inputs are not necessary.

Question 10

Question Type: MultipleChoice

Which option best is true about the summariesonly=t argument of the tstats command?

Options:

A- Applies only to accelerated data models.

B- When using an unaccelerated data model, the search produces a larger result count than with summariesonly=f.
C- Applies only to unaccelerated data models.
D- When using an accelerated data model, the search produces a larger result count than with summariesonly=f.

Answer:

A

Explanation:

Comprehensive and Detailed Step by Step

The summariesonly=t argument of the tstats command applies only to accelerated data models . It ensures that the search uses only the precomputed summaries of the data model, ignoring raw data.

Here's why this works:

Purpose of summariesonly=t : When set to true, the tstats command restricts the search to use only the accelerated summaries of the data model. This improves performance but may exclude events that are not part of the summary.

Accelerated Data Models : Acceleration creates summaries of data models, making them faster to query. Using summariesonly=t ensures that only these summaries are queried, avoiding raw data entirely.

Other options explained:

Option B : Incorrect because summariesonly=t does not apply to unaccelerated data models; it requires acceleration to function.

Option C : Incorrect because summariesonly=t applies only to accelerated data models, not unaccelerated ones.

Option D : Incorrect because summariesonly=t typically produces fewer results, as it excludes raw data that is not part of the summary.

Example:

| tstats count WHERE index=_internal summariesonly=t BY sourcetype

This query uses only the accelerated summaries of the _internal index.

Splunk Documentation on tstats:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/tstats

Splunk Documentation on Data Model Acceleration:

https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Acceleratedatamodels

P2P exams