PassQueen
a.k.a dumpsqueen.com

Splunk Core Certified Power User Exam

Splunk SPLK-1002

Version Demo

Total Demo Questions: 37

Total Premium Questions: 379

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Search and Reporting	154
Topic 2, Data Models	42
Topic 3, Pivot	10
Topic 4, Knowledge Objects	162
Topic 5, Creating Reports and Dashboards	11
Total	379

QUESTION NO: 1

Which of the following evaluation functions can be used with the eval command? (Choose all that apply.)

A. if()

B. case()

C. coalesce()

D. stats()

ANSWER: A B C

Explanation:

`if()`, `case()`, and `coalesce()` are evaluation functions that can be used in an `eval` expression. The `if()` function returns one value when a condition is true and another when it is false. The `case()` function evaluates conditions in order and returns the value associated with the first true condition. The `coalesce()` function returns the first value that is not null.

The `stats` command is not an evaluation function. It is a transforming command that calculates aggregate statistics over search results. See the Splunk evaluation functions reference.

QUESTION NO: 2

Which of the following statements describe saved reports? (Choose all that apply.)

A. A report can be scheduled to run automatically.

B. A report can be added to a dashboard as a panel.

C. A scheduled report can be configured to trigger alert actions.

D. Sharing a report automatically grants access to its indexes.

ANSWER: A B C

Explanation:

A saved report is a saved search that can be run again without recreating its SPL. Reports can be scheduled to run at defined intervals, and a scheduled report can be configured to trigger alert actions when its alert condition is met. Reports can also be added to dashboards as dashboard panels, allowing users to present saved search results and visualizations in a reusable dashboard layout.

Sharing a report does not automatically give all users access to the indexes searched by the report. Users must still have the appropriate index permissions to retrieve the underlying data. See the Splunk reports documentation and Splunk alerts documentation.

QUESTION NO: 3

In the Field Extractor, when would the regular expression method be used?

A. When events contain JSON data.

B. When events contain comma-separated data.

C. When events contain unstructured data.

D. When events contain table-based data.

ANSWER: C

Explanation:

When events contain unstructured data is correct because the Field Extractor's regular expression method is intended for event formats that do not have a consistent, simple delimiter separating every field. This is common in application logs, syslog-style messages, and free-form text, where values may be identified by surrounding labels, patterns, or variable spacing rather than by fixed columns. In Field Extractor, you can select a representative event and highlight values to extract; Splunk generates a proposed regular expression and provides tools to test and refine it against sample events. The resulting extraction identifies matching values in similarly formatted events and assigns them to fields for search and reporting. Regular expressions are therefore the appropriate method when extraction must describe a textual pattern instead of merely splitting an event at a common separator. Splunk supports creating these knowledge objects interactively and saving the extraction with the appropriate scope, such as a sourcetype, source, or host, so it can be applied consistently to relevant events. See Splunk's documentation on extracting fields interactively and its overview of Splunk regular expressions.

QUESTION NO: 4

Which search string would only return results for an event type called success ful_purchases?

A. tag=success ful_purchases

B. Event Type:: successful purchases

C. successful_purchases

D. event type—success ful_purchases

E. eventtype=successful_purchases

ANSWER: E

Explanation:

`eventtype=successful_purchases` is the appropriate search string because Splunk assigns the special `eventtype` field to events that match an event type definition. Searching that field with the event type's name restricts results to events classified by Splunk as belonging to `successful_purchases`. This is the field/value syntax Splunk documents for searching event types, and it is precise: the search is evaluated against the event type classification rather than simply looking for matching words in the raw event text.

Event type names commonly use underscores in place of spaces, which is also consistent with the intended name shown in the question. In SPL, field-value terms such as `eventtype=successful_purchases` are recognized search predicates. This makes the event type explicit and supports reliable filtering when the underlying events may contain many unrelated terms. Splunk's event-type documentation describes event types as named searches that categorize matching events and shows that the `eventtype` field can be used to search for those categorized events. See About event types and Search-time operations.

QUESTION NO: 5

Which of the following statements about tags is true? (select all that apply.)

A. Tags are case-insensitive.

B. Tags are based on field/value pairs.

C. Tags categorize events based on a search.

D. Tags are designed to make data more understandable.

ANSWER: B D

Explanation:

“Tags are based on field/value pairs” is correct because a Splunk tag is a knowledge object that associates a descriptive label with a field and one or more values for that field. For example, an administrator can associate the tag `authentication` with a particular `sourcetype`, host, or another extracted field value. When search results contain the defined field/value combination, Splunk can use that association to identify the relevant events. This lets searches use meaningful concepts through tag-based search syntax, without requiring users to repeatedly specify every underlying field/value criterion.

“Tags are designed to make data more understandable” is also correct. Tags add business- or operationally meaningful context to raw event data. A label such as `web`, `firewall`, or `privileged` can make related events easier for analysts to recognize, search, group, and report on, even when the original sources use differing technical field values. Splunk documents tags as a mechanism for grouping and identifying events that share common field/value characteristics, thereby improving the usability and interpretation of search data. See Splunk’s documentation on tags and aliases and adding and managing tags.

QUESTION NO: 6

Which of the following workflow actions can be executed from search results? (Choose all that apply.)

A. GET

B. POST

C. LOOKUP

D. Search

ANSWER: A B C D

Explanation:

Splunk workflow actions are context-sensitive actions made available from fields in search results. They enable users to take a value from an event or result and use it to initiate another operation without manually copying that value. The supported workflow-action types include **GET**, which opens a URL and passes field values as URL parameters; **POST**, which submits field values to an external endpoint; **Search**, which starts a secondary Splunk search using selected field values; and **LOOKUP**, which performs a lookup-related action using the selected value. These actions are configured by a Splunk administrator or knowledge-object owner and appear in the event or field context menus when their configured field and applicability conditions are met. They are especially useful for pivoting from events to related Splunk data, external ticketing or asset systems, and enrichment data. Splunk documents workflow actions as knowledge objects that can be defined for search-result fields and supports URL, search, and lookup-oriented workflows. See Splunk’s workflow action documentation and the workflow_actions.conf specification for configuration details.

QUESTION NO: 7

Why would the following search produce multiple transactions instead of one?

A. The maxspan option is not included.

B. The transaction command has a limit of 1000 events per transaction.

C. The transaction and commands cannot be used together.

D. The stats list () function is used.

ANSWER: B

Explanation:

The transaction command has a default maximum of 1,000 events per transaction. This behavior is controlled by the `maxevents` setting, whose default value is `1000`. When more than 1,000 events meet the transaction's grouping criteria, Splunk closes the current transaction after it reaches that limit and begins another transaction for subsequent matching events. Therefore, a search can return multiple transactions even when all events share the same specified field value and appear to belong to one logical activity. To allow a larger transaction, explicitly set `maxevents` to an appropriate value, such as `| transaction session_id maxevents=5000`. Use this carefully, since transaction processing can require substantial search resources when transactions contain many events. Splunk documents `maxevents` as the maximum number of events in a transaction and identifies 1,000 as its default value. See the Splunk transaction command reference and the optional transaction arguments documentation.

QUESTION NO: 8

When used with the timechart command, which value of the limit argument returns all values?

A. limit=*

B. limit=all

C. limit=none

D. limit=0

ANSWER: D

Explanation:

`limit=0` returns all series produced by a `timechart` split-by field. The `limit` argument controls how many split-by values, or series, Splunk retains in the results after ranking them according to the aggregation. Its default is 10, meaning that without an explicit setting, only the highest-ranking series are included. Setting the limit to zero disables that series limit, so Splunk does not filter out split-by values and returns every available value as a separate series. This setting is especially useful when the complete set of categories is needed for downstream analysis, reporting, or visualization rather than only the most prominent categories. The value is an integer argument, and zero has the documented special meaning of retaining all values. For the command syntax and the behavior of the `limit` option, see the Splunk Search Reference for timechart. Splunk also documents how time-based transforming commands generate and manage result series in its chart command reference.

QUESTION NO: 9

which of the following commands are used when creating visualizations(select all that apply.)

A. Geom

B. Choropleth

C. Geostats

D. iplocation

ANSWER: A C D

Explanation:

`geom`, `geostats`, and `iplocation` are SPL commands used to prepare or generate geographic data for map visualizations. The `geom` command adds geographical feature data from a KMZ or KML lookup to search results, allowing Splunk to render choropleth maps in which regions are colored according to a result value. This is useful when results contain a geographic identifier that matches features in the geographic lookup.

The `geostats` command computes statistical aggregations over geographic coordinates and formats the results for geographic visualizations, including cluster and marker-style maps. It can use latitude and longitude fields directly, or geographic fields derived earlier in the search.

The `iplocation` command enriches events containing IP addresses with geographic fields such as City, Country, latitude, and longitude. Those generated location fields can then be used by map visualizations and geographic search commands. Together, these commands support the workflow of deriving geographic context, aggregating events by location, and displaying the resulting data on a map. See Splunk's documentation for geom and the geostats command reference.

QUESTION NO: 10

For choropleth maps,splunk ships with the following KMZ files (select all that apply)

A. States of the United States

B. States and provinces of the united states and Canada

C. Countries of the European Union

D. Countries of the World

ANSWER: A D

Explanation:

Splunk includes geographic lookup definitions for choropleth visualizations covering **States of the United States** and **Countries of the World**. These supplied geographic datasets let a search associate result values with named geographic features and render each region with color intensity based on a measure. In SPL, the `geom` command performs this association by matching a location field in the search results to features in a KML or KMZ definition. For example, a search

containing state names or abbreviations can use the United States states geographic definition, while a search containing country names can use the world countries definition. This enables a choropleth without requiring an administrator to first create and deploy a custom geographic boundary file. Splunk documentation describes `geom` as the command for adding geographic feature information from KML/KMZ data to search results, and its mapping documentation covers choropleth maps and geographic data requirements. See the geom command reference and Splunk's choropleth map documentation.

QUESTION NO: 11

Which search commands allow a user to access data model summaries?

A. pivot, stats, and datamodel

B. pivot, tstats, and datamodel

C. transaction, tstats, and datamodel

D. stats, tstats, and datamodel

ANSWER: B

Explanation:

pivot, tstats, and datamodel is correct because these are the SPL commands designed to work with data models and their accelerated summaries. When a data model is accelerated, Splunk builds summary data for the model's eligible datasets over the configured acceleration range. This lets searches retrieve aggregate results from precomputed summaries rather than repeatedly processing all matching raw events, which can substantially improve performance for reporting and analytical use cases.

The `pivot` command runs a Pivot report against a data model and can use its acceleration. The `tstats` command performs statistical queries against indexed data and accelerated data-model summaries, using the `FROM datamodel=...` syntax to target a data model or dataset. The `datamodel` command provides search-time access to a data model's datasets and can use data-model acceleration when applicable. Together, these commands are the supported interfaces for accessing data-model summary information in SPL. See Splunk's Accelerate data models documentation and the tstats command reference.

QUESTION NO: 12

The eval command 'if' function requires the following three arguments (in order):

A. Boolean expression, result if true, result if false

B. Result if true, result if false, boolean expression

C. Result if false, result if true, boolean expression

D. Boolean expression, result if false, result if true

ANSWER: A

Explanation:

Boolean expression, result if true, result if false is correct because Splunk's `if` evaluation function uses the syntax `if(X, Y, Z)`. The first argument, `X`, is a Boolean expression that Splunk evaluates for each event or result. When that expression evaluates to true, the function returns the second argument, `Y`; when it evaluates to false, it returns the third argument, `Z`. For example, `eval status=if(bytes > 1000, "large", "small")` evaluates whether the `bytes` field exceeds 1000. Results meeting the condition receive the value `large`, and all remaining results receive `small`. This ordering enables conditional field creation and transformation directly in an `eval` expression, including within other SPL commands that accept evaluation expressions. Splunk documents this function explicitly as `if(X,Y,Z)`, where `X` is the

condition, `Y` is the value returned for true, and `Z` is the value returned for false. See the official Splunk conditional functions reference and the eval command reference.

QUESTION NO: 13

Which of the following statements describe the search below? (Choose all that apply.)

index=main | transaction clientip host maxspan=30s maxpause=5s

A. Events in the transaction occurred within 5 seconds.

B. It groups events that share the same clientip and host.

C. The first and last events are no more than 5 seconds apart.

D. The first and last events are no more than 30 seconds apart

ANSWER: B D

Explanation:

The search first retrieves events from the `main` index and then uses the `transaction` command to combine qualifying events into transactions. **It groups events that share the same clientip and host.** This is correct because the fields listed immediately after `transaction` are the transaction grouping fields. Events must have matching values for both `clientip` and `host` to be considered part of the same transaction.

The first and last events are no more than 30 seconds apart is also correct. The `maxspan=30s` setting limits a transaction's total duration: the time difference between its earliest and latest event cannot exceed 30 seconds. The search additionally includes `maxpause=5s`, which limits the allowable time gap between adjacent events in a transaction. That setting helps define transaction continuity, while `maxspan` defines the full earliest-to-latest time range. Splunk documents both settings as transaction constraints applied while grouping events. See the Splunk transaction command reference and the Splunk Search Reference: transaction.

QUESTION NO: 14

Sally created several tags for employees of Buttercup Games. She tagged each employee's badge number with the department name and location. Which search query would Sally use to filter for employees of the Marketing department who do not work in the San_Francisco office?

A. tag!=Marketing tag=San_Francisco

B. tag=Marketing NOT (tag=San_Francisco)

C. tag=Marketing exclude (tag=San_Francisco)

D. tag::Marketing!=San_Francisco

ANSWER: B

Explanation:

The correct search is `tag=Marketing NOT (tag=San_Francisco)`. In Splunk, tags are searchable through the `tag` field using the syntax `tag=<tag_name>`. Specifying `tag=Marketing` limits the matching events to those associated with the Marketing tag. The Boolean `NOT` operator then removes any events that also have the San_Francisco tag, leaving only Marketing employees whose tagged location is not San_Francisco. Parentheses clearly group the location-tag condition being excluded, which is especially useful as searches become more complex. This use of tags enables knowledge objects to add meaningful, searchable classifications to event data without requiring the original event fields to be changed. Splunk's search documentation describes Boolean expressions, including `NOT`, for excluding matching terms, and its knowledge-manager documentation explains how tags can be used as searchable field/value annotations. See the Splunk Boolean search reference and Splunk documentation on tags and aliases.

QUESTION NO: 15

When using multiple expressions in a single eval command, which delimiter is used?

A. , (comma)

B. I (pipe)

C. / (forward slash)

D. : (colon)

ANSWER: A

Explanation:

The correct delimiter is **, (comma)**. In Splunk Search Processing Language, the `eval` command can calculate and assign more than one field within the same command invocation. Each field-assignment expression is written in the form `field_name=expression`, and separate assignments are joined with commas. For example, `| eval full_name=first_name." ".last_name, is_error=if(status>=500, "true", "false")` creates both `full_name` and `is_error` in one `eval` command. This approach keeps related calculations together and avoids adding unnecessary pipeline stages to a search. The comma is part of the `eval` command syntax for a list of expressions; each expression is evaluated against the events reaching that point in the search pipeline. Splunk documents this syntax as `eval [field=<expression> ...]`, with multiple expressions separated by commas in practical search usage. See the Splunk eval command reference and the Search Reference for SPL command syntax and usage.

QUESTION NO: 16

Which delimiters can the Field Extractor (FX) detect? (select all that apply)

A. Tabs

B. Pipes

C. Spaces

D. Commas

ANSWER: B C D

Explanation:

The Field Extractor can automatically identify commonly occurring delimiter characters in representative event data and offer delimiter-based extraction when the event structure supports it. Pipes, spaces, and commas are among the delimiter patterns that FX can detect for this workflow. After a delimiter is selected, FX uses it to split the event into field/value segments, lets the user assign field names to the resulting columns, and saves the extraction as a knowledge object when appropriate. This approach is particularly useful for consistently structured events such as pipe-delimited application records, space-separated log entries, and comma-separated values. Splunk documents delimiter-based extraction as one of the methods available in the Field Extractor and describes how FX detects delimiter candidates from the sample event before the user validates the field layout. For further details, see Splunk's Field Extractor method-selection documentation and the interactive field extraction documentation.

QUESTION NO: 17

Which of the following statements about tags is true? (Choose all that apply.)

A. Tags are case-insensitive.

B. Tags are based on field/value pairs.

C. Tags categorize events based on a search.

D. Tags are designed to make data more understandable.

ANSWER: B D

Explanation:

“Tags are based on field/value pairs” is correct because a Splunk tag is a knowledge object that associates one or more descriptive labels with a particular field and value combination. For example, an administrator can tag `host=webserver01` with a meaningful label such as `production`. When events contain the matching field/value pair, Splunk applies the tag at search time, allowing users to locate or group related events more easily. Tags can be applied to extracted fields, including default fields such as `host`, `source`, and `sourcetype`, as well as custom fields.

“Tags are designed to make data more understandable” is also correct. Tags provide business- or operationally meaningful labels that make raw event metadata easier to interpret and use in searches, reports, dashboards, and data models. Rather than requiring every user to know each underlying field value, a consistent tag can express a shared category or concept across relevant events. This supports clearer searches, such as using the `tag` field to retrieve events associated with a designated label. Splunk documents tags as a way to classify events using field-value pairs and to improve the organization and usability of search results. See Splunk: About tags and aliases and Splunk: Tag event types and saved searches.

QUESTION NO: 18

Data models are composed of one or more of which of the following datasets? (select all that apply)

A. Transaction datasets

B. Events datasets

C. Search datasets

D. Any child of event, transaction, and search datasets

ANSWER: A B C D

Explanation:

Splunk data models organize knowledge about data into hierarchical datasets. A data model can contain one or more root datasets and can extend those roots with child datasets, allowing the model to represent increasingly specific subsets of the data. The supported root dataset types are transaction datasets, event datasets, and search datasets. Transaction datasets define transactions by grouping related events; event datasets represent events that meet specified constraints; and search datasets are based on the results of a search. A child dataset can be created beneath an event, transaction, or search dataset and inherits the parent dataset’s constraints while adding further constraints or fields. Therefore, transaction datasets, event datasets, search datasets, and children of those dataset types can all be components of a data model. This hierarchy is particularly useful for pivoting and for maintaining a consistent, reusable structure for data analysis. Splunk’s documentation describes data models as hierarchies of datasets and identifies event, search, transaction, and child datasets as the available dataset types. See the Splunk documentation on data models and the Splunk Splexicon definition of a data model dataset.

QUESTION NO: 19

What happens when a user edits the regular expression (regex) field extraction generated in the Field Extractor (FX)?

A. There is a limit to the number of fields that can be extracted.

B. The user is unable to preview the extractions.

C. The extraction is added at index time.

D. The user is unable to return to the automatic field extraction workflow.

ANSWER: D

Explanation:

The user is unable to return to the automatic field extraction workflow. Splunk's Field Extractor initially provides an interactive, guided workflow that generates a regular expression from the fields and sample events selected by the user. If the user directly edits that generated regular expression, the extraction becomes a manually edited regex. This is a one-way transition: Splunk no longer retains the guided extraction settings in a form that can be reopened and adjusted through the automatic workflow. The user can continue working with the regex-based extraction and save it as a search-time field extraction, but subsequent changes must be made by editing the regular expression rather than returning to the guided field-selection process. This behavior matters when designing extractions, because users should complete as much of the automatic workflow as possible before making direct regex modifications. Splunk documents that field extractions created through the Field Extractor are search-time knowledge objects and describes the interactive extraction workflow and its regex editing behavior in its knowledge-management documentation. See Splunk Docs: Extract fields interactively with the Field Extractor and Splunk Docs: Create field extractions.

QUESTION NO: 20

Which of the following statements about the spath command are true? (Choose all that apply.)

A. It can extract fields from JSON data.

B. It can extract fields from XML data.

C. It can use the input argument to specify a field other than _raw.

D. It permanently adds extracted values to indexed events.

ANSWER: A B C

Explanation:

The `spath` command extracts fields from structured XML and JSON data. By default, it reads structured data from the `_raw` field, but the `input` argument can identify another field that contains the XML or JSON data. The `path` argument can be used to extract a specific element or value from the structured data.

The command is a search-time extraction command. It does not modify indexed raw data. See the Splunk spath command reference.

QUESTION NO: 21

The timechart command is an example of which of the following command types?

A. Orchestrating

B. Transforming

C. Statistical

D. Generating

ANSWER: B

Explanation:

Transforming is correct because `timechart` takes the events returned by a search and transforms them into an aggregated, time-based results table suitable for charting. It groups events into time bins and applies statistical functions, such as `count`, `avg`, or `sum`, to produce a series of summarized values over time. For example, `index=main | timechart count` converts individual matching events into counts for successive time intervals. A split-by field can also create separate series for distinct field values. This behavior matches Splunk's definition of a transforming command: it

transforms event-level search results into a structured table of statistics, which can then be visualized as a chart. Splunk's command reference explicitly categorizes `timechart` as a transforming command and describes it as creating a time-series chart and corresponding statistics table. See the timechart command reference and Splunk's search command types documentation.

QUESTION NO: 22

What are the two parts of a root event dataset?

A. Fields and variables.

B. Fields and attributes.

C. Constraints and fields.

D. Constraints and lookups.

ANSWER: C

Explanation:

A root event dataset is defined by **constraints and fields**. In a Splunk data model, the root event dataset establishes the base set of events that belong to that branch of the model. Its constraints are search conditions that determine event membership, such as a specific sourcetype, source, host, index, or field/value condition. Splunk applies these constraints to identify the events represented by the dataset.

The dataset's fields then define the useful extracted information that the data model exposes for those matching events. A field can be based on an existing event field or calculated through an eval expression, allowing the data model to provide a consistent, reusable schema for searches, reports, pivots, and downstream dataset objects. Together, the constraints establish the event population and the fields describe the information available from that population. This is the fundamental structure of an event dataset at the root of a data-model hierarchy. Splunk's data-model documentation describes event datasets as being composed of constraints and fields, and explains how root datasets provide the initial event set for a data model. See Splunk: About data models and Splunk: Define data model objects.

QUESTION NO: 23

These users can create global knowledge objects. (Select all that apply.)

A. users

B. power users

C. administrators

ANSWER: B C

Explanation:

Power users and administrators can create global knowledge objects when their roles have the applicable knowledge-object editing and sharing capabilities. Global sharing makes a knowledge object available across all apps, subject to its permissions, so it is appropriate for reusable items such as event types, field extractions, tags, workflow actions, and saved searches that need to be available broadly. The default Power User role is intended for advanced search users who create and share knowledge objects, while administrators have full administrative authority over knowledge objects and their permissions. In Splunk, role capabilities ultimately control what a user may create, edit, and share; therefore, an administrator can customize a role's permissions if an organization needs a different governance model. For the standard role expectations tested in Splunk Core Power User material, both power users and administrators are authorized to create globally shared knowledge objects. See Splunk's documentation on managing knowledge object permissions and its reference for roles and capabilities.

QUESTION NO: 24

Which of the following can be used with the eval command tostring function (select all that apply)

A. ‘‘hex’’

B. ‘‘commas’’

C. ‘‘Decimal’’

D. ‘‘duration’’

ANSWER: A B D

Explanation:

The `tostring` evaluation function converts a numeric value into a string, and its optional format parameter controls how that resulting string is rendered. `''hex''` is a supported format and represents the numeric value as a hexadecimal string. This is useful when values need to be displayed or compared in base-16 notation, such as identifiers or bit-oriented values.

`''commas''` is also supported. It formats a number with thousands separators, making large numeric results easier for people to read in search output, dashboards, and reports. For example, a value such as 1234567 can be rendered as 1,234,567.

`''duration''` is a supported format for converting a numeric duration, expressed in seconds, into Splunk’s readable duration representation. This is particularly useful when presenting elapsed times or other interval values in search results. The format argument is supplied as the second parameter, for example: `eval readable_time=tostring(seconds, "duration")`.

Splunk documents the available `tostring` formats, including `commas`, `duration`, and `hex`, in its Search Reference: Conversion functions. Additional syntax and evaluation-function context is available in the Common eval functions reference.

QUESTION NO: 25

Which knowledge object is used to normalize field names to comply with the Splunk Common Information Model (CIM)?

A. Field alias

B. Event types

C. Search workflow action

D. Tags

ANSWER: A

Explanation:

Field alias is used to normalize field names for Common Information Model compliance. A field alias creates an alternate name for an existing extracted field at search time without changing the underlying raw event data. This is particularly useful when different products or data sources represent the same concept with different field names. For example, a source-specific field can be aliased to the CIM-defined field name expected for its data model, allowing searches, dashboards, and apps to work consistently across normalized sources.

Splunk describes field aliases as a method for mapping one field name to another at search time. In CIM implementations, this supports mapping vendor- or source-specific fields to the common field names defined by the applicable CIM data model. The original field remains available, while the common alias provides the standardized schema needed for cross-source correlation and CIM-aware content. For implementation guidance, see Splunk’s field alias documentation and the Splunk Common Information Model documentation.

QUESTION NO: 26

Which of the following statements describes POST workflow actions?

A. Configuration of a POST workflow action includes choosing a sourcetype.

B. POST workflow actions can be configured to send email to the URI location.

C. By default, POST workflow actions are shown in both the event and field menus.

D. POST workflow actions can be configured to send POST arguments to the URI location.

ANSWER: D

Explanation:

POST workflow actions can be configured to send POST arguments to the URI location. In Splunk, a workflow action is a knowledge object that provides an interactive link or action from event or field context menus. A POST workflow action sends an HTTP POST request to a specified URI when a user invokes it. Its configuration includes POST arguments, which are name/value pairs sent in the body of that request. Argument values can use field tokens, allowing Splunk to substitute values from the selected event or field dynamically before the request is submitted. This lets analysts pass relevant context, such as an IP address, user name, host, or other extracted field value, to an external web application, ticketing system, lookup service, or investigation tool. The URI identifies the destination endpoint, while the POST arguments define the data delivered to that endpoint. This behavior is specifically what distinguishes the POST workflow action configuration: it supports sending contextual argument data through an HTTP POST operation. Splunk documents the URI and POST-argument settings in its workflow action configuration guidance: Set up a POST workflow action. For broader context on workflow actions as knowledge objects and how they are invoked from search results, see About workflow actions.

QUESTION NO: 27

Which of the following describes this search?

New Search

'third_party_outages(EMEA,-24h)'

A. This search will find all events for the third_party_outages event type that have "EMEA" or "-24h" in the raw event data.

B. This search will run the third_party_outages saved search and filter for events containing "EMEA" and "-24h" in the raw event data.

C. This search will run the third_party_outages macro and pass the arguments EMEA and -24h to the macro definition.

D. This search will find all events in the third_party_outages index with the tags EMEA and -24h.

ANSWER: C

Explanation:

This search will run the third_party_outages macro and pass the arguments EMEA and -24h to the macro definition. In Splunk, a search macro is a reusable, named segment of SPL that is expanded into its underlying search definition when it is invoked. Macros can be defined with arguments, allowing the same reusable SPL logic to accept values supplied by the user at search time. The macro invocation syntax uses the macro name followed by its argument list in parentheses; here, `third_party_outages` receives two positional arguments: `EMEA` and `-24h`. How those values are used depends on the macro definition, such as substituting them into search terms, field constraints, or time-related logic. The enclosing single quotation marks identify the expression as a search macro invocation, rather than a literal keyword search. This makes macros useful for standardizing common searches and reducing duplicated SPL across users and dashboards. Splunk expands the macro before executing the resulting SPL. See Splunk's documentation on defining search macros and search-time syntax and modifiers.

QUESTION NO: 28

Which workflow action type performs a secondary search?

A. POST

B. Drilldown

C. GET

D. Search

ANSWER: D

Explanation:

Search is the workflow action type that performs a secondary search. In Splunk, workflow actions are knowledge objects that let users take action on field values displayed in search results. A Search workflow action constructs and runs another Splunk search when a user selects the action, substituting one or more values from the selected event into the secondary search string. This supports contextual investigation without requiring the user to manually copy a value and build a follow-up query. For example, selecting an action for an IP address field can launch a search for other events containing that IP address during a relevant time range. Search workflow actions are configured with a search string and can use field-value tokens so the resulting secondary search is dynamically tailored to the event from which it was invoked. Splunk documents Search alongside GET and POST as the workflow action types available in Splunk Web, and specifically describes Search actions as a way to run secondary searches based on event field values. See the Splunk documentation on defining workflow actions and workflow action concepts.

QUESTION NO: 29

Which of the following is true about data model attributes?

A. They cannot be created within the data model.

B. They can only be added into a root search dataset.

C. They cannot be edited if inherited from a parent dataset.

D. They can be added to a dataset from search time field extractions.

ANSWER: C

Explanation:

They cannot be edited if inherited from a parent dataset. Data model datasets can inherit fields, including attributes, from their parent datasets. This inheritance helps maintain a consistent field definition across related datasets and avoids duplicating configuration. An inherited attribute is controlled by the dataset where it was originally defined, so its definition is read-only in child datasets. To change details such as the attribute's field name or display name, the data model editor requires the change to be made in the parent dataset that owns the attribute. The updated definition is then inherited by applicable child datasets. This behavior preserves the structure and semantics of the data model hierarchy, ensuring that a child dataset continues to use the parent's shared field definition rather than silently creating a divergent version. Splunk's data model documentation describes how fields are inherited through dataset hierarchies and how field definitions are managed in the data model editor. See Splunk documentation: About data models and Splunk documentation: Data model fields.

QUESTION NO: 30

Which Knowledge Object does the Splunk Common Information Model (CIM) use to normalize data, in addition to field aliases, event types, and tags?

A. Macros

B. Lookups

C. Workflow actions

D. Field extractions

ANSWER: B D

Explanation:

The correct answers are **Lookups** and **Field extractions**. CIM normalization is primarily performed at search time by applying Splunk knowledge objects that make source-specific data conform to CIM's common field names and expected values. Field extractions identify and extract values from raw event text so that data is available as searchable fields. This is essential when a source does not already provide the fields required by a CIM data model or when the relevant value must be parsed from an event payload.

Lookups also support normalization by mapping existing field values to standardized values or by adding contextual fields from an external mapping table. For example, a lookup can translate vendor-specific action, status, or category values into values expected by a CIM field. Together with field aliases, event types, and tags, these knowledge objects enable disparate data sources to be searched consistently and accelerated through CIM data models. Splunk's CIM documentation describes using search-time field extractions and lookups as part of the normalization process: Use the CIM to normalize data at search time. Splunk also documents how lookups add fields and map values during searches: About lookups.

QUESTION NO: 31

What functionality does the Splunk Common Information Model (CIM) rely on to normalize fields with different names?

A. Macros.

B. Field aliases.

C. The rename command.

D. CIM does not work with different names for the same field.

ANSWER: B

Explanation:

Field aliases. are the appropriate functionality because CIM normalization requires data from varied products, vendors, and sourcetypes to be represented using a consistent set of field names. A field alias maps an existing source-specific field name to an alternate name without changing the original field or its value. For example, a data source that provides a source-specific user field can be mapped to the CIM-defined `user` field. Searches, data models, dashboards, and apps that use CIM can then work against the common CIM field name rather than requiring separate search logic for every source format.

Field aliases are knowledge objects configured through `fields.conf` or Splunk Web. They are applied at search time, which preserves the raw event and original extracted fields while making the normalized field available for use. This approach supports the CIM objective of providing a shared semantic model across different data sources and enables CIM data models to accelerate consistent analysis. Splunk documents field aliases as a way to assign alternate names to fields, and CIM documentation describes normalizing source data to the fields expected by its data models. See Use field aliases and Overview of the Splunk Common Information Model.

QUESTION NO: 32

Which of the following statements about the stats command are true? (Choose all that apply.)

A. It can calculate aggregate functions such as count and avg.

B. It can group aggregate results by field values.

C. It returns a statistics table instead of the individual raw events.

D. It creates transactions from related events.

ANSWER: A B C

Explanation:

The `stats` command calculates aggregate statistics over search results. It can calculate functions such as `count`, `sum`, `avg`, `min`, and `max`. A `by` clause groups the statistics by one or more field values, such as `| stats count by host`.

The command is transforming, so it returns a statistics table rather than preserving the individual raw events that were used to calculate the results. The `stats` command does not create transactions; the `transaction` command is used to group related events into transaction results. See the Splunk stats command reference.

QUESTION NO: 33

Which of the following Statements about macros is true? (select all that apply)

A. Arguments are defined at execution time.

B. Arguments are defined when the macro is created.

C. Argument values are used to resolve the search string at execution time.

D. Argument values are used to resolve the search string when the macro is created.

ANSWER: B C

Explanation:

“Arguments are defined when the macro is created” is correct because a Splunk search macro can be configured with argument names as part of its definition. These named arguments act as placeholders in the macro’s definition, allowing the same reusable search logic to accept different inputs. For example, a macro author can define an argument such as `index`, `sourcetype`, or a threshold value, then use that argument within the macro’s search string. This structure is established when the knowledge object is created or edited.

“Argument values are used to resolve the search string at execution time” is also correct. When a user invokes an argument-based macro in a search, Splunk substitutes the supplied argument values into the macro definition as the search is parsed and run. This makes macros useful for standardizing recurring SPL while retaining flexibility for search-specific values. Splunk describes search macros as reusable chunks of SPL and documents that macros may be defined with arguments, which are specified when the macro is called. See Splunk’s About search macros and Define search macros.

QUESTION NO: 34

Which of the following searches show a valid use of a macro? (Choose all that apply.)

A. index=main source=mySource oldField=* |'makeMyField(oldField)'| table _time newField

B. index=main source=mySource oldField=* | stats if('makeMyField(oldField)') | table _time newField

C. index=main source=mySource oldField=* | eval newField='makeMyField(oldField)'| table _time newField

D. index=main source=mySource oldField=* | "'newField('makeMyField(oldField)')'" | table _time newField

ANSWER: A C

Explanation:

`index=main source=mySource oldField=* |'makeMyField(oldField)'| table _time newField` is a valid macro-use pattern when the macro expands to a complete search command, such as an `eval` expression that creates `newField`. Search macros are reusable SPL fragments that are expanded into the search before Splunk processes the resulting SPL. Consequently, a macro can encapsulate an entire command pipeline element and receive a field name as an argument.

`index=main source=mySource oldField=* | eval newField='makeMyField(oldField)'| table _time newField` is also a valid macro-use pattern when the macro expands to an expression appropriate for the right-hand side of `eval`. Parameterized macros permit callers to supply values such as `oldField`, allowing the macro definition to apply a reusable expression to that field. In actual SPL, macro references are enclosed in backticks; the rendered apostrophe-style delimiters in these choices represent the macro reference notation. The expansion must produce syntactically valid SPL in its surrounding context, whether that context is a standalone command or an `eval` expression. Splunk documents macro definitions, arguments, and invocation syntax in Define search macros; the broader guidance for reusable knowledge objects is available in the Splunk Knowledge Manager manual.

QUESTION NO: 35

Given the event below, how can the value in the Zip_Code field be used to retrieve the local weather from an external resource?

25/Oct/2023:20:29:43 , 151.131.173.143 , V2.003 , Zip_Code: 75890 , DataCenter: DC1

A. Create a POST workflow action.

B. Create a GET workflow action.

C. Create a PUT workflow action.

D. Create a Search workflow action.

ANSWER: B

Explanation:

Create a GET workflow action. In Splunk, workflow actions let users take action on field values displayed in search results. A GET workflow action is designed to construct and open a URL, substituting a field value such as `Zip_Code` into the URL or its query string. For example, an administrator can configure a weather-service URL whose location parameter is populated with the selected event's `Zip_Code` value. When a user invokes the action for this event, Splunk resolves the field value `75890` and sends the browser to the external weather resource, where that value can be used to retrieve weather data for the associated location. This is the standard workflow-action pattern for navigating from a Splunk event to an external lookup or information service using a value extracted from that event. The action can be configured in Splunk Web under workflow actions, with field tokens included in the URI so the target is dynamically tailored to the selected result. Splunk's documentation describes GET workflow actions as URL-based actions and documents how workflow actions use field values in their configuration. See Configure workflow actions and Use workflow actions.

QUESTION NO: 36

Use the dedup command to _____.

A. Rename a field in the index

B. remove duplicate values

C. provide an additional alias for the field that can D.be used in the search criteria

ANSWER: B

Explanation:

The `dedup` command removes duplicate events from search results based on one or more specified fields, retaining the first event encountered for each distinct field value by default. For example, `... | dedup host` returns only one event per host, while `... | dedup user src_ip` retains one event for each unique combination of user and source IP address. This is useful when search results contain repeated events or when a report should show a single representative event for each value. The command operates on events in their current search order, so using it after sorting can determine which event is retained. Splunk's command reference describes `dedup` as removing subsequent results that match a previous result on the specified field or fields. Therefore, "remove duplicate values" is the best completion of the statement. See the Splunk dedup command reference and the Splunk Search Reference.

QUESTION NO: 37

When you mouse over and click to add a search term this (thesE. Boolean operator(s) is(arE. not implied. (Select all that apply).

A. OR

B. ( )

C. AND

D. NOT

ANSWER: A B D

Explanation:

OR, **( )**, and **NOT** are correct because they express search logic that must be explicitly entered in an SPL search. Selecting a field value from the Fields sidebar or clicking a value in an event adds that value as a search term; it does not infer that the newly added term should be an alternative, a negation, or part of a deliberately grouped expression. To search for either of multiple terms, the search must contain the uppercase `OR` operator. To exclude matching events, the search must contain `NOT` (or an equivalent explicit exclusion expression). Parentheses must likewise be typed when grouping is needed to control the order in which combined Boolean conditions are evaluated. This is particularly important for searches that mix alternative and exclusion logic, where intended grouping needs to be visible in the search string. Splunk documents its Boolean-search syntax, including uppercase `OR`, `NOT`, and parentheses for grouping, in the Boolean expressions reference. Splunk also explains interactive field and event actions in its search-results interaction documentation.