PER SCHOLAS

Lesson 200.4 Search Language Fundamentals

Learning Objectives

At the end of this lesson, learners will be able to:

- Describe the search pipeline.
- Use the following SPL basic commands:
  - The fields command.
  - The dedup command.
  - The sort command.
  - The eval command.

Introduction

SPL (Search Processing Language) is the search language used in Splunk for querying and analyzing data.

It is a proprietary language specifically designed for interacting with data in Splunk.

SPL provides a powerful and flexible set of commands, functions, and operators that allow users to search, filter, transform, and visualize data in Splunk.

It supports various data processing tasks such as data retrieval, field extraction, filtering, aggregation, statistical analysis, and more.

In this lesson, we will describe the search pipeline and explore some basic SPL commands.

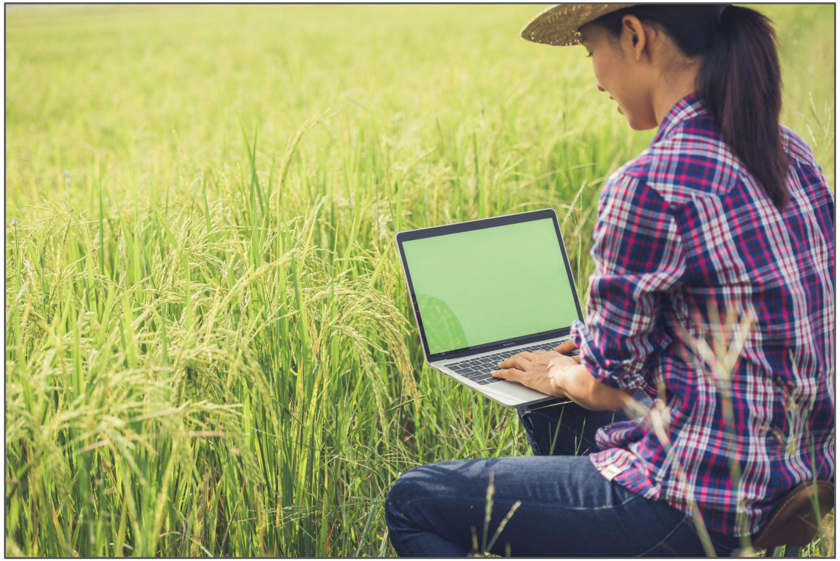


image: Freepik.com

4.1 Describe the Search Pipeline

Anatomy of a search.

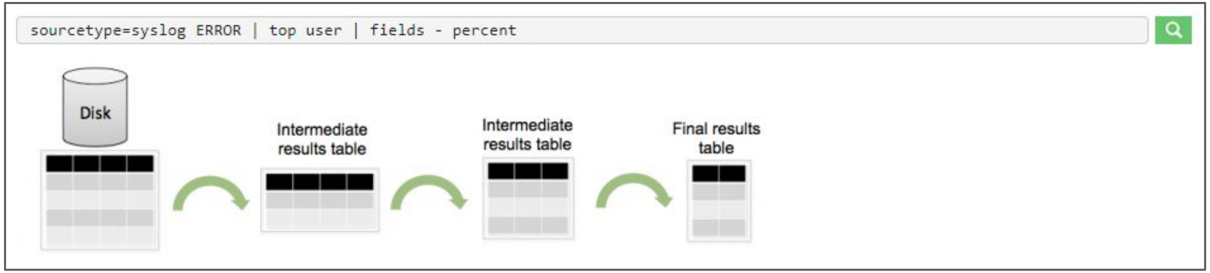
- A search consists of a **series of commands** that are delimited by **pipe (|)** characters.
- The first **whitespace-delimited string** after each pipe character controls the command used.
- The remainder of the text for each command is handled in a manner **specific** to the **given command**.

source: https://docs.splunk.com/Documentation/Splunk/9.0.4/Search/Aboutsearchlanguagesyntax

4.1 Describe the Search Pipeline

Anatomy of a search.

- For example, let's take a look at the following search.



```
sourcetype=syslog ERROR | top user | fields - percent
```

We will look at the **top** and **fields** commands in more detail later on in this lesson.

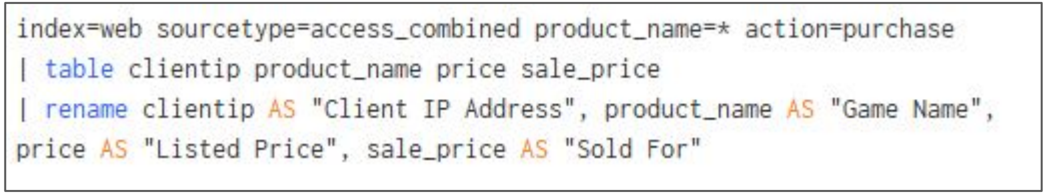
- The Disk represents **all** of your **indexed data**.
- The **first intermediate** results table shows events retrieved from the index that matched the search terms "**sourcetype=syslog ERROR.**"
- The **second intermediate** results table shows the results of the **top** command, "**top user,**" summarizing the events into a list of the top 10 users, the user count, and percentage.
- The "**fields - percent**" removes the column that shows the **percentage;** therefore, you are left with a smaller final results table.

4.1 Describe the Search Pipeline (continued)

- The **search pipeline** in Splunk refers to the **sequential execution of commands** that process and transform data during a search operation.
- It involves **chaining together** multiple **commands** using the **pipe symbol (|)** to pass the **results** from one **command** to the **next**.
- Each command in the pipeline performs a **specific operation** on the data, such as filtering, statistical analyzing, aggregating, or visualizing it.
- When you perform a search in Splunk, the **initial search** command **retrieves events** or data based on your specified **criteria**.
- The **results** of this command then **serve as the input** for **subsequent commands** in the pipeline.
- Each **command** takes the **output** from the **previous command**, processes it further, and generates a **new set of results**.

4.1 Describe the Search Pipeline (continued)

- A Splunk search **starts with search terms** at the **beginning** of the **pipeline**. These search terms are **keywords**, **phrases**, **boolean expressions**, **key/value pairs**, etc. that specify which **events** you want to **retrieve** from the index(es).
- The retrieved events can then be **passed as inputs** into a search **command** using a **pipe** character. **Search commands** tell Splunk software what to do to the events **after you retrieved** them from the index(es)
- In this example, the **search results** - **events** from the **web index** with the **access_combined** sourcetype, that contain **any value** for the key **produvt_name**, and contain the value **purchase** for the key **action** - are **piped** into the **table** command, and the **output** of the table command is **piped** into the **rename** command.



```
index=web sourcetype=access_combined product_name=* action=purchase
| table clientip product_name price sale_price
| rename clientip AS "Client IP Address", product_name AS "Game Name",
price AS "Listed Price", sale_price AS "Sold For"
```

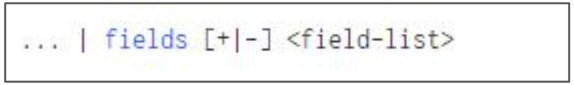
image: screenshot, splunk Search & Reporting app

4.1 Describe the Search Pipeline - Summary

- The Splunk search pipeline refers to the sequential flow of data processing and transformation in a search.
- The pipe symbol | is used to separate individual commands within the search pipeline, allowing you to apply various operations and transformations to your data.
- When using the pipe symbol, the output of one command becomes the input for the next command in the pipeline.
- This enables you to chain multiple commands together to perform complex data manipulations, filtering, aggregation, and analysis.

4.2 Use Basic SPL Commands

The fields command

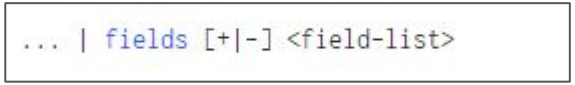


```
... | fields [+|-] <field-list>
```

- The **fields** command in Splunk is used to **include** or **exclude** specific **fields** in the search results.
- **... | fields + <field-list>** will **include only** the **specified fields** in the results.
- Using the fields command **without** a [+|-] sign will **default to [+]**.
- **... | fields - <field-list>** will **include all** fields **except** the **specified fields** in the results.
- The fields command **supports wildcards** in the **<field-list>** argument.
- By default, the **internal** fields **_raw** and **_time** are **included** in the output.

4.2 Use Basic SPL Commands (continued)

The fields command:



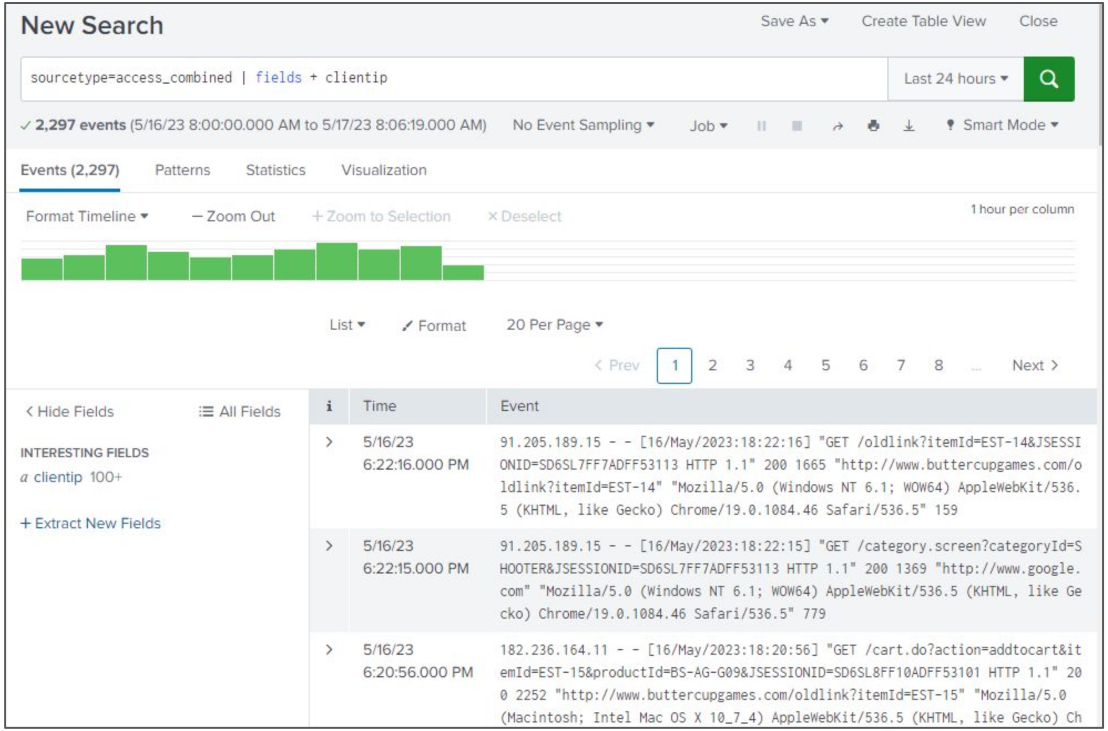
```
... | fields [+|-] <field-list>
```

- **Extracting fields** from data **during a search** is a **resource-intensive** process that consumes **significant computational resources** and time.
- **field-list** is a **required** argument. Other arguments are **optional**.
- To **improve performance**, use the **fields** or **fields +** command **immediately** after the **basic search string**.
- To **remove** fields form view on **statistics** and **visualizations** use the **fields -** command at the end of the search string.

4.2 Use Basic SPL Commands (continued)

The fields command

- Review the **results** of the following usage of the **fields** command **displayed** in the **image**.
- Note that the **only field included** in the fields sidebar is **clientip**.



```
sourcetype=access_combined | fields + clientip
```

✓ 2,297 events (5/16/23 8:00:00.000 AM to 5/17/23 8:06:19.000 AM)

INTERESTING FIELDS

a clientip 100+

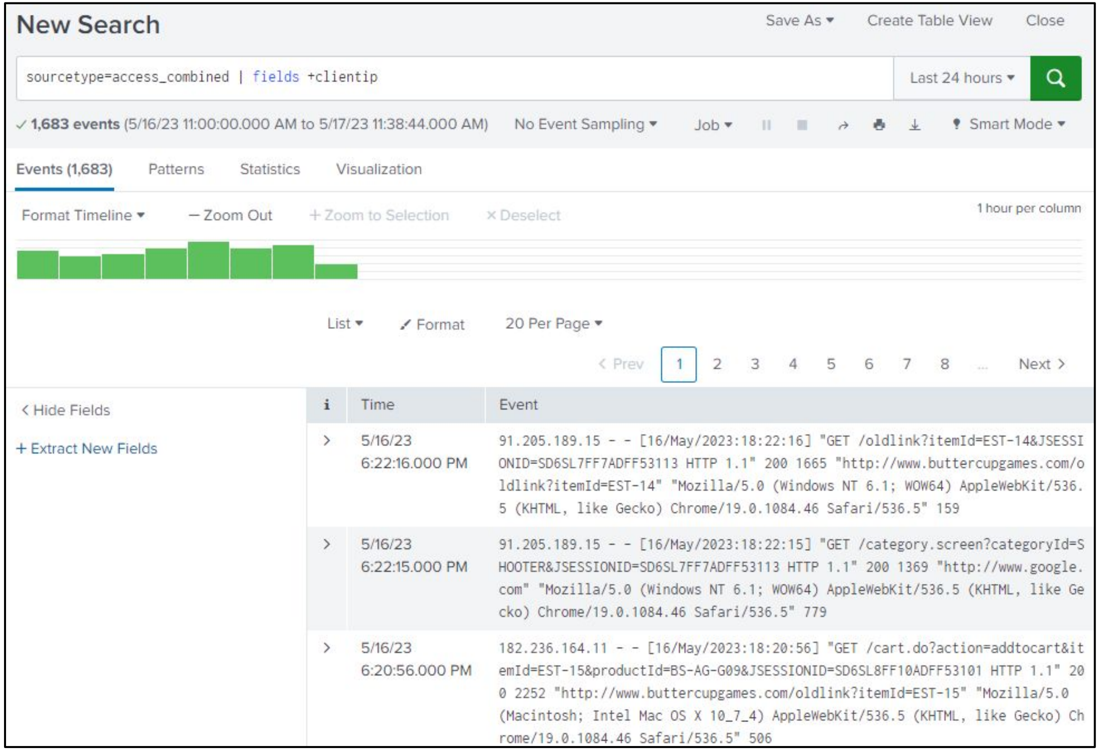
i	Time	Event
›	5/16/23 6:22:16.000 PM	91.205.189.15 - - [16/May/2023:18:22:16] "GET /oldlink?itemId=EST-14&JSESSIONID=SD6SL7FF7ADFF53113 HTTP 1.1" 200 1665 "http://www.buttercupgames.com/oldlink?itemId=EST-14" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 159
›	5/16/23 6:22:15.000 PM	91.205.189.15 - - [16/May/2023:18:22:15] "GET /category.screen?categoryId=SHOOTER&JSESSIONID=SD6SL7FF7ADFF53113 HTTP 1.1" 200 1369 "http://www.google.com" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 779
›	5/16/23 6:20:56.000 PM	182.236.164.11 - - [16/May/2023:18:20:56] "GET /cart.do?action=addtocart&itemId=EST-15&productId=BS-AG-G09&JSESSIONID=SD6SL8FF10ADFF53101 HTTP 1.1" 200 2252 "http://www.buttercupgames.com/oldlink?itemId=EST-15" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Ch

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The fields command

- Ensure to **leave a space** between the **[+]** sign and the **<fields-list>**.
- Make sure you specify **at least one field** name in the <field-list> argument.
- Falling to do so, will **exclude all fields** for the search results.



```
sourcetype=access_combined | fields +clientip
```

✓ 1,683 events (5/16/23 11:00:00.000 AM to 5/17/23 11:38:44.000 AM)

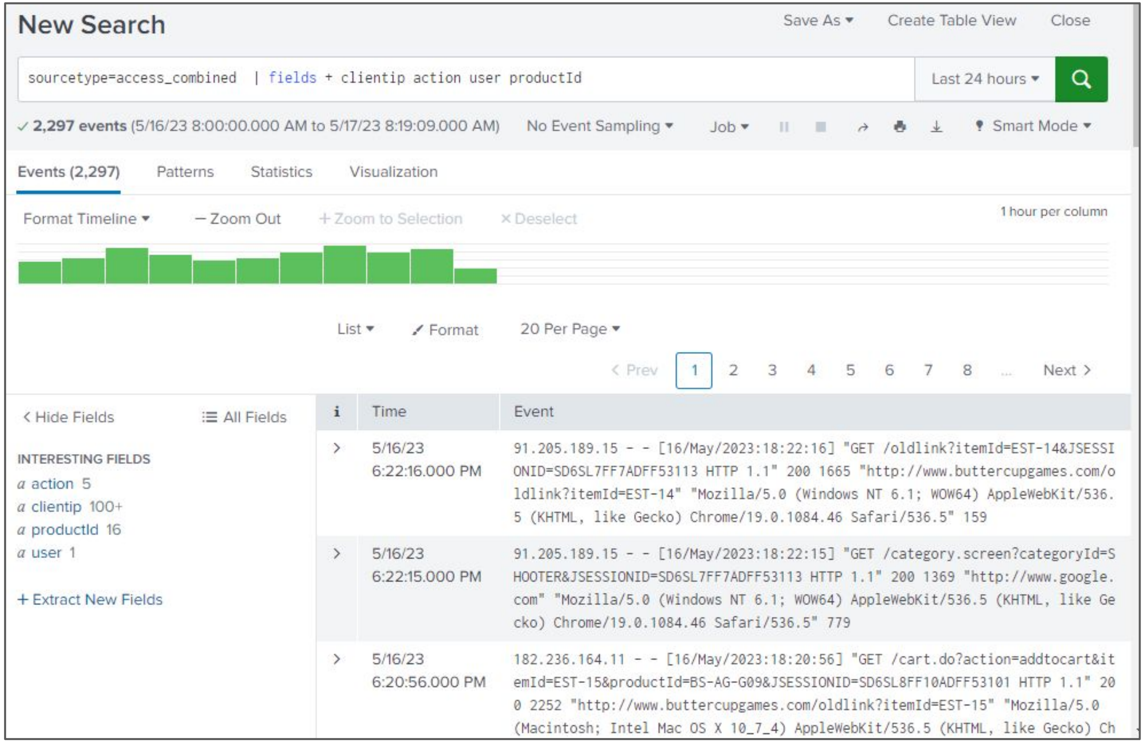
i	Time	Event
›	5/16/23 6:22:16.000 PM	91.205.189.15 - - [16/May/2023:18:22:16] "GET /oldlink?itemId=EST-14&JSESSIONID=SD6SL7FF7ADFF53113 HTTP 1.1" 200 1665 "http://www.buttercupgames.com/oldlink?itemId=EST-14" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 159
›	5/16/23 6:22:15.000 PM	91.205.189.15 - - [16/May/2023:18:22:15] "GET /category.screen?categoryId=SHOOTER&JSESSIONID=SD6SL7FF7ADFF53113 HTTP 1.1" 200 1369 "http://www.google.com" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 779
›	5/16/23 6:20:56.000 PM	182.236.164.11 - - [16/May/2023:18:20:56] "GET /cart.do?action=addtocart&itemId=EST-15&productId=BS-AG-G09&JSESSIONID=SD6SL8FF10ADFF53101 HTTP 1.1" 200 2252 "http://www.buttercupgames.com/oldlink?itemId=EST-15" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 506

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The fields command

- To **include more than one** field using the fields command, add **more fields** to the <fields-list> argument **separated** by a **space**.



New Search

```
sourcetype=access_combined  | fields + clientip action user productId
```

✓ **2,297 events** (5/16/23 8:00:00.000 AM to 5/17/23 8:19:09.000 AM)

INTERESTING FIELDS
- a action 5
- a clientip 100+
- a productId 16
- a user 1

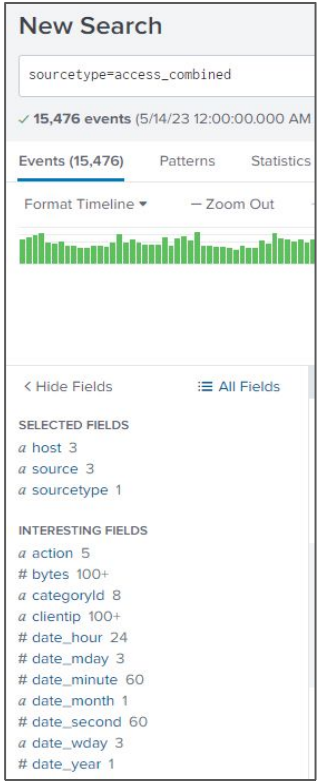
i	Time	Event
›	5/16/23 6:22:16.000 PM	91.205.189.15 - - [16/May/2023:18:22:16] "GET /oldlink?itemId=EST-14&JSESSIONID=SD6SL7FF7ADFF53113 HTTP 1.1" 200 1665 "http://www.buttercupgames.com/oldlink?itemId=EST-14" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 159
›	5/16/23 6:22:15.000 PM	91.205.189.15 - - [16/May/2023:18:22:15] "GET /category.screen?categoryId=SHOOTER&JSESSIONID=SD6SL7FF7ADFF53113 HTTP 1.1" 200 1369 "http://www.google.com" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 779
›	5/16/23 6:20:56.000 PM	182.236.164.11 - - [16/May/2023:18:20:56] "GET /cart.do?action=addtocart&itemId=EST-15&productId=BS-AG-G09&JSESSIONID=SD6SL8FF10ADFF53101 HTTP 1.1" 200 2252 "http://www.buttercupgames.com/oldlink?itemId=EST-15" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Ch

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The fields command

- Compare the **results** of the following usage of the **fields** command **displayed** in the **images**.
- Note that the **only field missing** in the fields sidebar is **categoryId**.

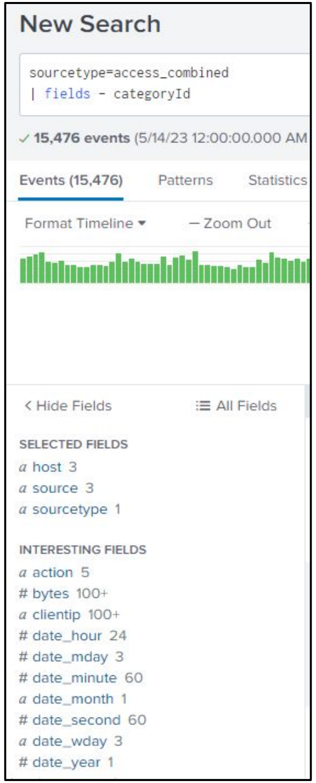


```
sourcetype=access_combined
```

✓ 15,476 events (5/14/23 12:00:00.000 AM

SELECTED FIELDS
a host 3
a source 3
a sourcetype 1

INTERESTING FIELDS
a action 5
# bytes 100+
a categoryId 8
a clientip 100+
# date_hour 24
# date_mday 3
# date_minute 60
a date_month 1
# date_second 60
a date_wday 3
# date_year 1



```
sourcetype=access_combined
| fields - categoryId
```

✓ 15,476 events (5/14/23 12:00:00.000 AM

SELECTED FIELDS
a host 3
a source 3
a sourcetype 1

INTERESTING FIELDS
a action 5
# bytes 100+
a clientip 100+
# date_hour 24
# date_mday 3
# date_minute 60
a date_month 1
# date_second 60
a date_wday 3
# date_year 1

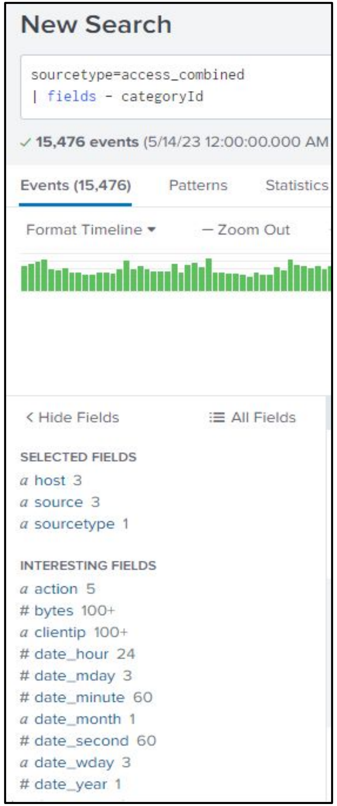
image: screenshot, splunk Search & Reporting app

PER SCHOLAS

4.2 Use Basic SPL Commands (continued)

The fields command

- Compare the **results** of the following usage of the **fields** command **displayed** in the **images**.
- Note that **omitting** the **space** between the **-** (minus) sign and the **field name** will result in the **removal of all fields from the search**.



New Search

```
sourcetype=access_combined
| fields - categoryId
```

✓ 15,476 events (5/14/23 12:00:00.000 AM

Events (15,476) Patterns Statistics

Format Timeline ▾ — Zoom Out

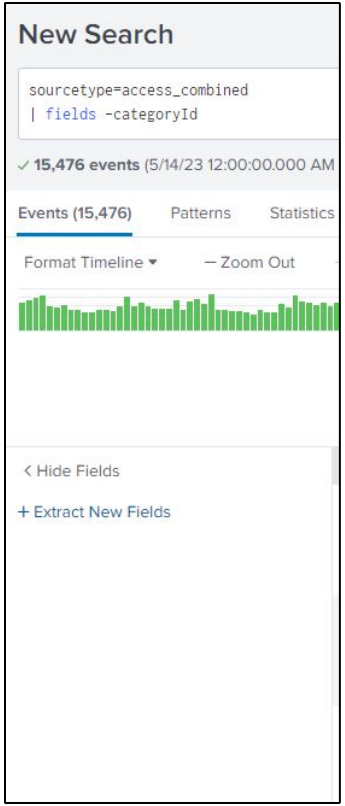
< Hide Fields ☰ All Fields

SELECTED FIELDS

a host 3
a source 3
a sourcetype 1

INTERESTING FIELDS

a action 5
# bytes 100+
a clientip 100+
# date_hour 24
# date_mday 3
# date_minute 60
a date_month 1
# date_second 60
a date_wday 3
# date_year 1



New Search

```
sourcetype=access_combined
| fields -categoryId
```

✓ 15,476 events (5/14/23 12:00:00.000 AM

Events (15,476) Patterns Statistics

Format Timeline ▾ — Zoom Out

< Hide Fields

+ Extract New Fields

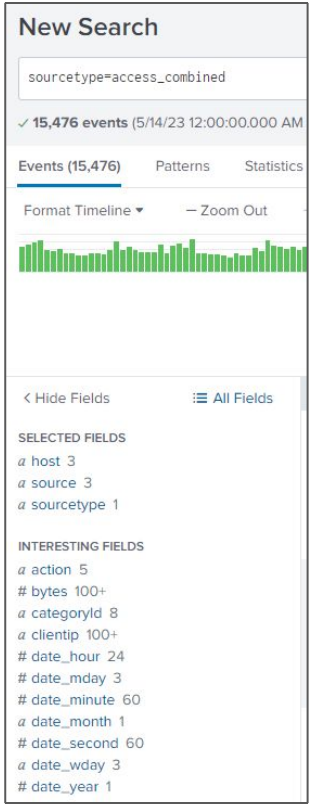
image: screenshot, splunk Search & Reporting app

PER SCHOLAS

4.2 Use Basic SPL Commands (continued)

The fields command

- To **exclude more than one** field using the fields command, add **more fields** to the <fields-list> argument **separated** by a **space**.
- Compare the **results** of the following usage of the **fields** command **displayed** in the **images**.



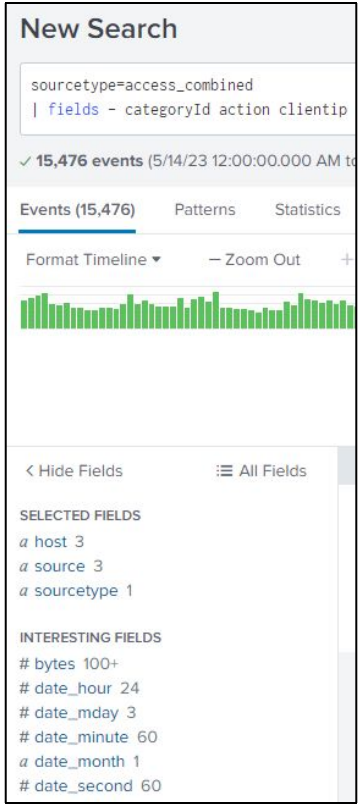
New Search

```
sourcetype=access_combined
```

✓ 15,476 events (5/14/23 12:00:00.000 AM

SELECTED FIELDS
a host 3
a source 3
a sourcetype 1

INTERESTING FIELDS
a action 5
# bytes 100+
a categoryId 8
a clientip 100+
# date_hour 24
# date_mday 3
# date_minute 60
a date_month 1
# date_second 60
a date_wday 3
# date_year 1



New Search

```
sourcetype=access_combined
| fields - categoryId action clientip
```

✓ 15,476 events (5/14/23 12:00:00.000 AM t

SELECTED FIELDS
a host 3
a source 3
a sourcetype 1

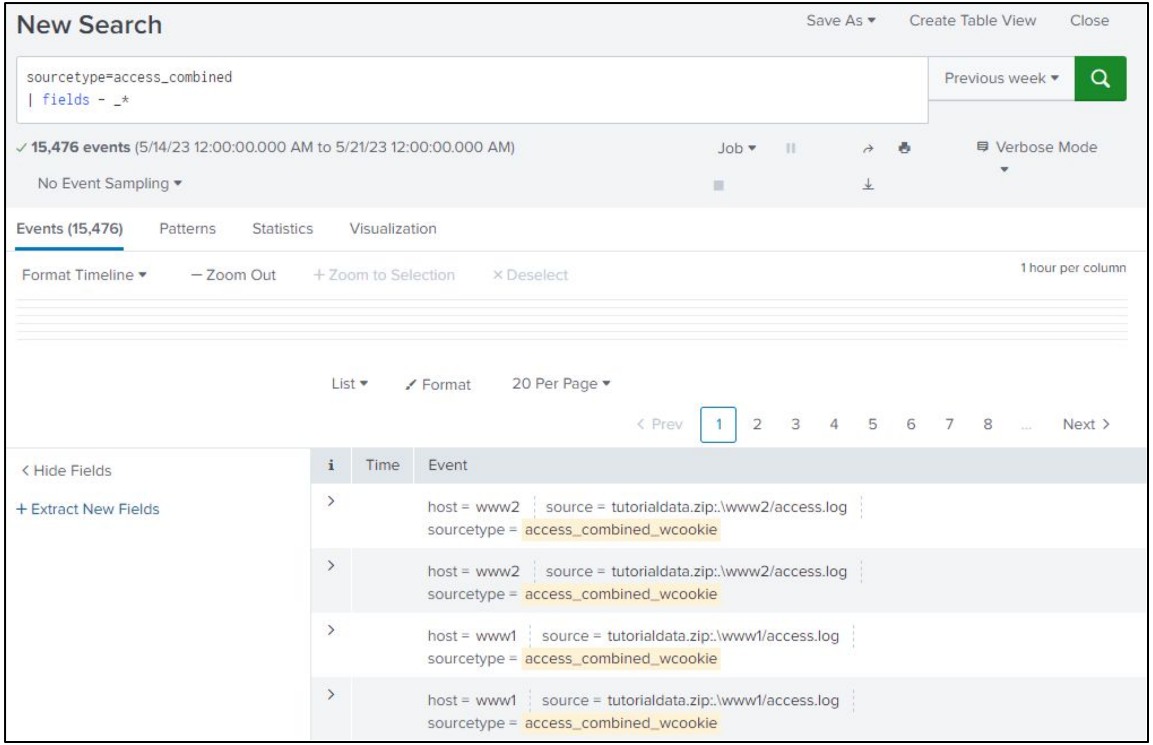
INTERESTING FIELDS
# bytes 100+
# date_hour 24
# date_mday 3
# date_minute 60
a date_month 1
# date_second 60

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The fields command

- To **exclude internal fields** use **fields - _raw _time**
- To **exclude all internal fields** using a wildcard …| **fields - _***



New Search

```
sourcetype=access_combined
| fields - _*
```

✓ 15,476 events (5/14/23 12:00:00.000 AM to 5/21/23 12:00:00.000 AM)

No Event Sampling

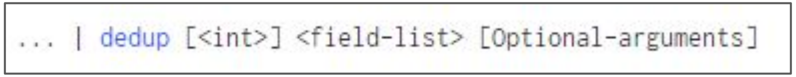
Verbose Mode

i	Time	Event
>		host = www2 source = tutorialdata.zip:.\www2/access.log sourcetype = access_combined_wcookie
>		host = www2 source = tutorialdata.zip:.\www2/access.log sourcetype = access_combined_wcookie
>		host = www1 source = tutorialdata.zip:.\www1/access.log sourcetype = access_combined_wcookie
>		host = www1 source = tutorialdata.zip:.\www1/access.log sourcetype = access_combined_wcookie

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The dedup command

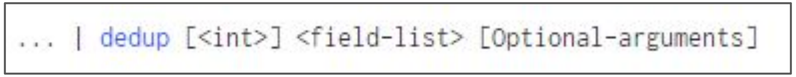


```
... | dedup [<int>] <field-list> [Optional-arguments]
```

- The **dedup** command in Splunk **removes** the events that contain an i**dentical combination of values** for the **fields** that you **specify**.
- **field-list** is a **required** argument. Other arguments are **optional**.
- You can specify the **number of events with duplicate values** to keep. You can also **sort** the fields, which determines which event is retained.
- Other options enable you to **retain events with the duplicate fields removed**, or to **keep** events where the fields specified **do not exist in the events**.
- Events **returned** by dedup are **based on search order**. For historical searches, the **most recent events** are searched first. For real-time searches, the **first events that are received** are searched, which are **not necessarily** the most recent events.

4.2 Use Basic SPL Commands (continued)

The dedup command



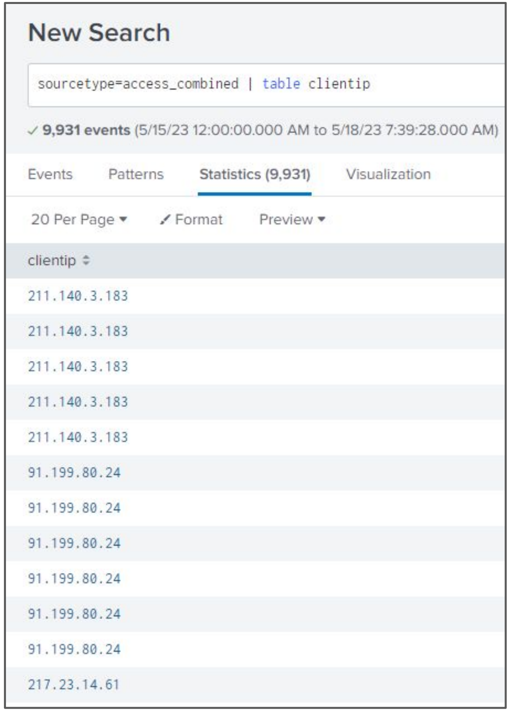
```
... | dedup [<int>] <field-list> [Optional-arguments]
```

- The dedup command is a **streaming command** by **default** or a **dataset processing command**, depending on which **arguments** are specified with the command.
- A **streaming command** operates on **each event** as the event is **returned** by a search.
- A **dataset processing command** is a command that **requires** the **entire dataset before** the command can run.
- If you specify the **<sort-by-clause> argument**, the dedup command acts as a **dataset processing command**. All of the **results** must be **collected** before **sorting**.
- **Avoid** using the dedup command on the **_raw** field if you are searching over a **large volume** of data. The text of every event in memory is retained which **impacts** your search **performance**.

4.2 Use Basic SPL Commands (continued)

The dedup command - Examples

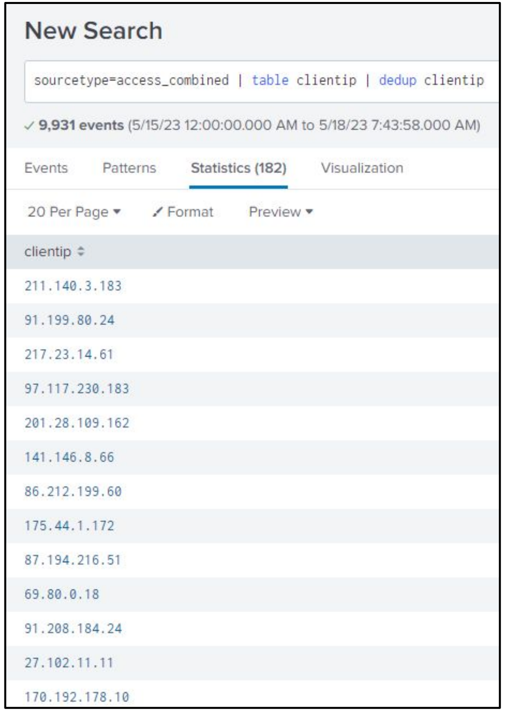
- Use **dedup** to clear **duplicate clientip** addresses from a table.
- Note the **different number** of the table entries under the **Statistics** tab.



```
sourcetype=access_combined | table clientip
```

✓ **9,931 events** (5/15/23 12:00:00.000 AM to 5/18/23 7:39:28.000 AM)

clientip ⇕
211.140.3.183
211.140.3.183
211.140.3.183
211.140.3.183
211.140.3.183
91.199.80.24
91.199.80.24
91.199.80.24
91.199.80.24
91.199.80.24
91.199.80.24
217.23.14.61



```
sourcetype=access_combined | table clientip | dedup clientip
```

✓ **9,931 events** (5/15/23 12:00:00.000 AM to 5/18/23 7:43:58.000 AM)

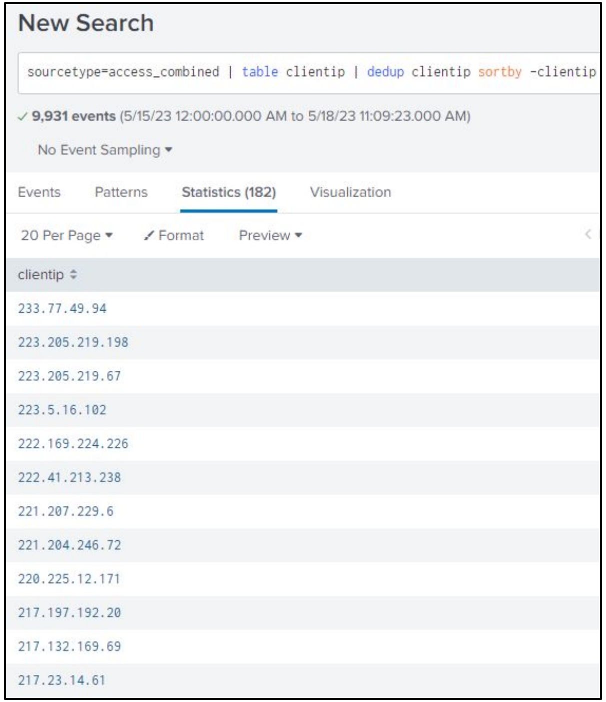
clientip ⇕
211.140.3.183
91.199.80.24
217.23.14.61
97.117.230.183
201.28.109.162
141.146.8.66
86.212.199.60
175.44.1.172
87.194.216.51
69.80.0.18
91.208.184.24
27.102.11.11
170.192.178.10

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL commands.

The dedup command - Examples

- Use **dedup** with the **sortby** option to sort results in **ascending** or **descending** order.
- Note how the **+** and **-** sign **affect** the **results**.

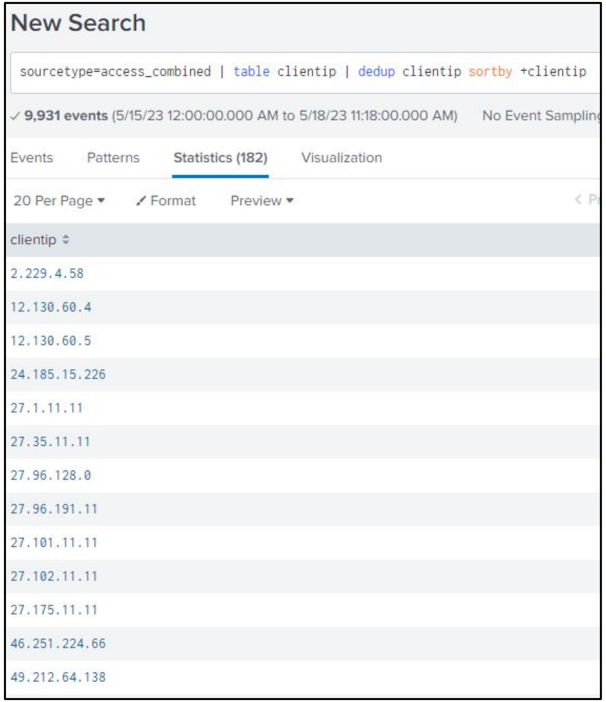


New Search

```
sourcetype=access_combined | table clientip | dedup clientip sortby -clientip
```

✓ 9,931 events (5/15/23 12:00:00.000 AM to 5/18/23 11:09:23.000 AM)

clientip ⇕
233.77.49.94
223.205.219.198
223.205.219.67
223.5.16.102
222.169.224.226
222.41.213.238
221.207.229.6
221.204.246.72
220.225.12.171
217.197.192.20
217.132.169.69
217.23.14.61



New Search

```
sourcetype=access_combined | table clientip | dedup clientip sortby +clientip
```

✓ 9,931 events (5/15/23 12:00:00.000 AM to 5/18/23 11:18:00.000 AM)

clientip ⇕
2.229.4.58
12.130.60.4
12.130.60.5
24.185.15.226
27.1.11.11
27.35.11.11
27.96.128.0
27.96.191.11
27.101.11.11
27.102.11.11
27.175.11.11
46.251.224.66
49.212.64.138

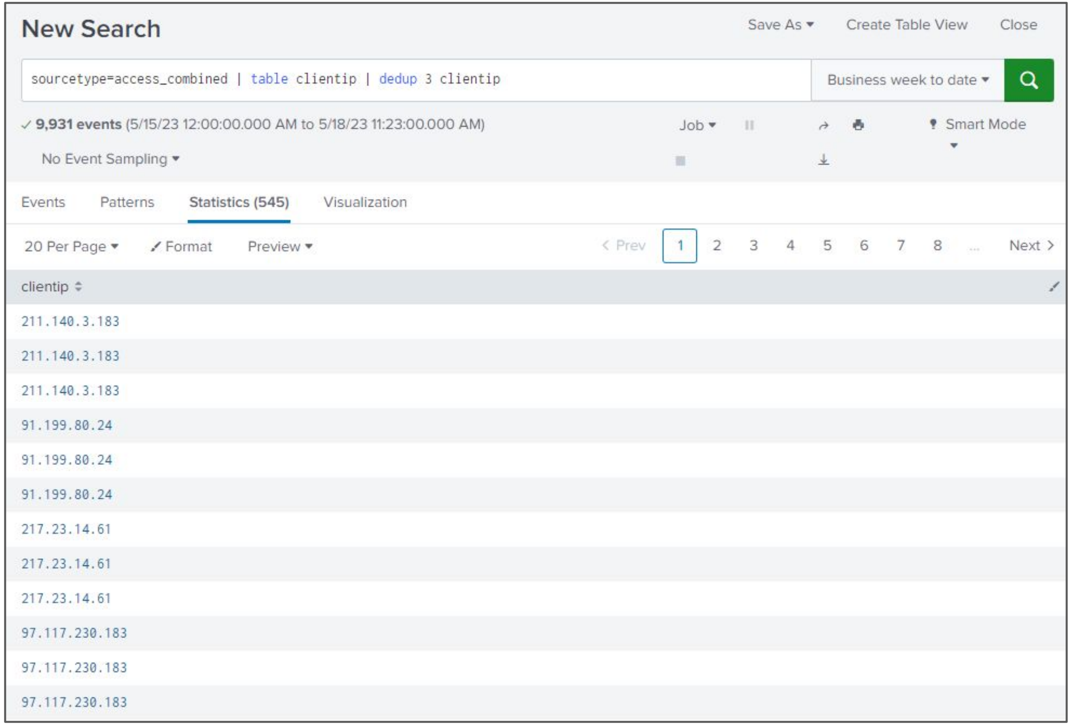
Note: Sorting results by IP addresses is probably not very useful, except for this demonstration.

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL commands.

The dedup command - Examples

- Keep the **first 3 duplicate** results by adding the number **3** as an **argument** to dedup **before** specifying the field-list.
- Note that **more options** are **available** for the dedup command. Refer to the Splunk documentation for more details.



```
sourcetype=access_combined | table clientip | dedup 3 clientip
```

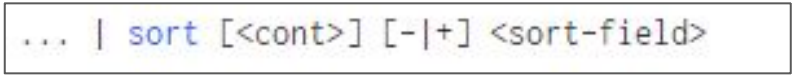
✓ 9,931 events (5/15/23 12:00:00.000 AM to 5/18/23 11:23:00.000 AM)

clientip
211.140.3.183
211.140.3.183
211.140.3.183
91.199.80.24
91.199.80.24
91.199.80.24
217.23.14.61
217.23.14.61
217.23.14.61
97.117.230.183
97.117.230.183
97.117.230.183

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The sort command

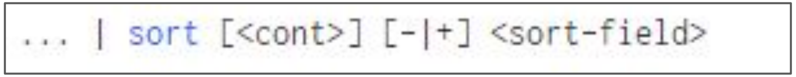


```
... | sort [<cont>] [-|+] <sort-field>
```

- The sort command **sorts** all of the **results** by the **specified fields**.
- **sort-field** is a **required** argument. Other arguments are **optional**.
- Results **missing** a given field are treated as having the **smallest** or **largest** possible value of that field if the order is **descending** or **ascending**, respectively.
- If the first **argument** to the **sort** command is a **number**, then at **most,** that many results are **returned**, in order. If **no number** is specified, the default limit of **10000** is used. If the number **0** is specified, **all of the results** are returned.
- Use a **minus** sign (-) for **descending** order and a **plus** sign (+) for **ascending** order.
- When specifying **more** than one **field**, **separate** the field names with **commas**.

4.2 Use Basic SPL commands.

The sort command



```
... | sort [<cont>] [-|+] <sort-field>
```

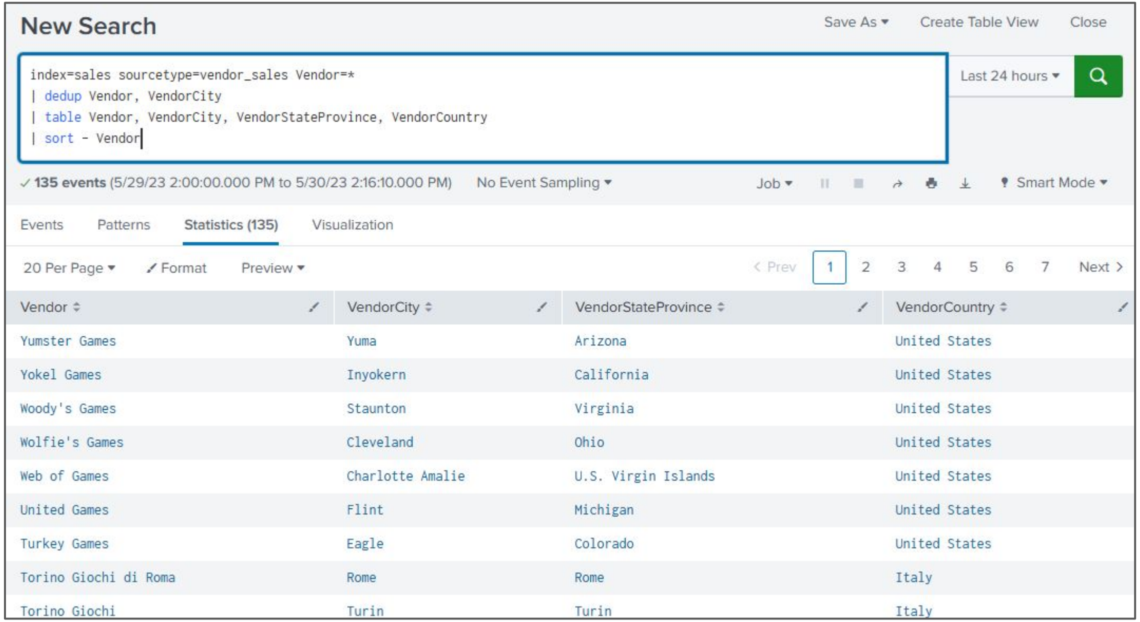
- Sorting operation is based on the field type.
  - Alphanumeric strings are sorted in **lexicographic** order.
    - **lexicographic** order means **uppercase** letters appear **before lowercase** letters.
  - Numeric fields are sorted numerically.
  - Include a space after +/-

Unsorted	Lexicographical sort
cat	AND
Bee	Bee
ant	Foo
AND	ant
Foo	bar
bar	cat

4.2 Use Basic SPL Commands (continued)

The sort command - examples.

- Display vendor information; sort in **descending** order.



```
index=sales sourcetype=vendor_sales Vendor=*
| dedup Vendor, VendorCity
| table Vendor, VendorCity, VendorStateProvince, VendorCountry
| sort - Vendor
```

✓ 135 events (5/29/23 2:00:00.000 PM to 5/30/23 2:16:10.000 PM) No Event Sampling ▾

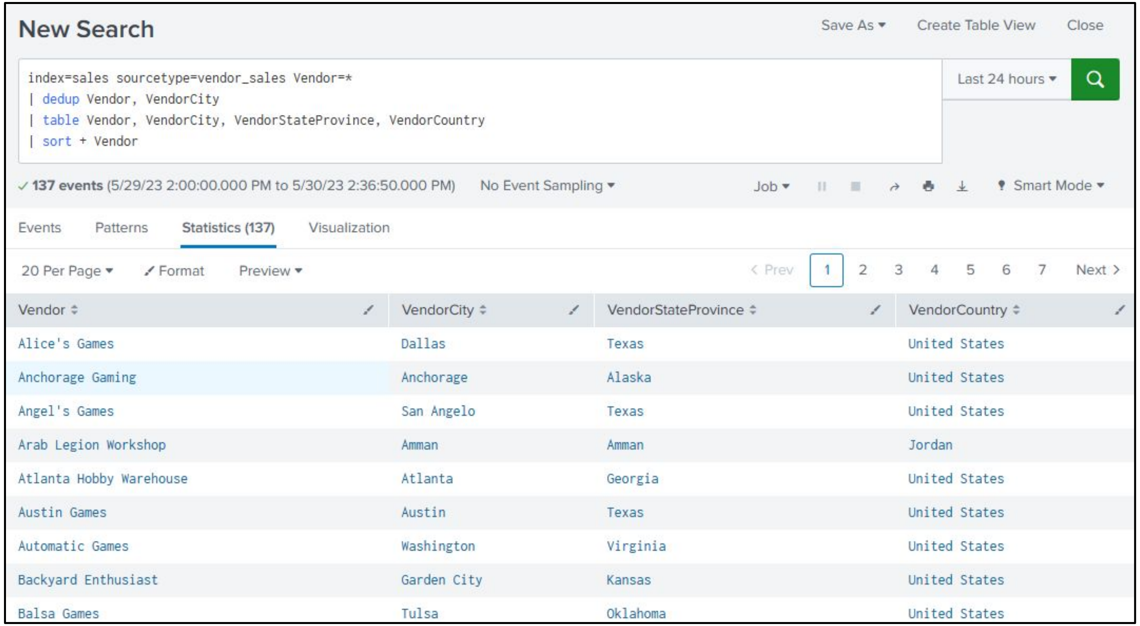
Vendor	VendorCity	VendorStateProvince	VendorCountry
Yumster Games	Yuma	Arizona	United States
Yokel Games	Inyokern	California	United States
Woody's Games	Staunton	Virginia	United States
Wolfie's Games	Cleveland	Ohio	United States
Web of Games	Charlotte Amalie	U.S. Virgin Islands	United States
United Games	Flint	Michigan	United States
Turkey Games	Eagle	Colorado	United States
Torino Giochi di Roma	Rome	Rome	Italy
Torino Giochi	Turin	Turin	Italy

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The sort command - examples.

- Display vendor information; sort in **ascending** order.



```
index=sales sourcetype=vendor_sales Vendor=*
| dedup Vendor, VendorCity
| table Vendor, VendorCity, VendorStateProvince, VendorCountry
| sort + Vendor
```

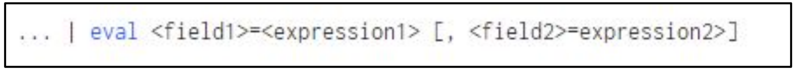
✓ 137 events (5/29/23 2:00:00.000 PM to 5/30/23 2:36:50.000 PM) No Event Sampling ▾

Vendor	VendorCity	VendorStateProvince	VendorCountry
Alice's Games	Dallas	Texas	United States
Anchorage Gaming	Anchorage	Alaska	United States
Angel's Games	San Angelo	Texas	United States
Arab Legion Workshop	Amman	Amman	Jordan
Atlanta Hobby Warehouse	Atlanta	Georgia	United States
Austin Games	Austin	Texas	United States
Automatic Games	Washington	Virginia	United States
Backyard Enthusiast	Garden City	Kansas	United States
Balsa Games	Tulsa	Oklahoma	United States

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The eval command



```
... | eval <field1>=<expression1> [, <field2>=expression2>]
```

The **eval** command **calculates an expression** and puts the **resulting value** into a **search results field**.

- It is extremely **powerful** and supports a vast assortment of functions for performing specific tasks.
- The eval command supports various **operators**.
- You can chain **multiple eval expressions** in one search using a comma to separate subsequent expressions. The search processes multiple eval expressions **left-to-right.** and lets you reference **previously evaluated** fields in subsequent expressions.

PER SCHOLAS

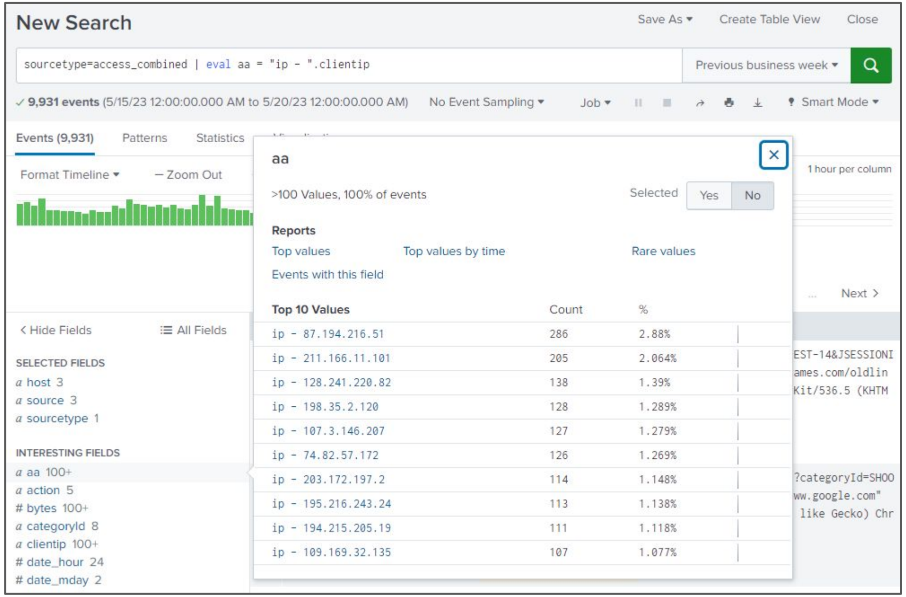
4.2 Use Basic SPL Commands (continued)

The eval command

```
... | eval <field1>=<expression1> [, <field2>=expression2>]
```

The **eval** command **calculates an expression** and puts the **resulting value** into a **search results field**.

- If the **field name** that you specify does **not match** a field in the **output**, a **new field** is **added** to the **search results**, but **nothing** is **added** to the **index**.
- In this example, “ip - “ is appended using the dot (.) operator to the existing values in the clientip field and stored in a new field named aa



```
sourcetype=access_combined | eval aa = "ip - ".clientip
```

✓ 9,931 events (5/15/23 12:00:00.000 AM to 5/20/23 12:00:00.000 AM)

>100 Values, 100% of events

Top 10 Values	Count	%
ip - 87.194.216.51	286	2.88%
ip - 211.166.11.101	205	2.064%
ip - 128.241.220.82	138	1.39%
ip - 198.35.2.120	128	1.289%
ip - 107.3.146.207	127	1.279%
ip - 74.82.57.172	126	1.269%
ip - 203.172.197.2	114	1.148%
ip - 195.216.243.24	113	1.138%
ip - 194.215.205.19	111	1.118%
ip - 109.169.32.135	107	1.077%

SELECTED FIELDS

a host 3
a source 3
a sourcetype 1

INTERESTING FIELDS

a aa 100+
a action 5
# bytes 100+
a categoryId 8
a clientip 100+
# date_hour 24
# date_mday 2

```
EST-14&JSESSIONI
ames.com/oldlin
Kit/536.5 (KHTM

?categoryId=SHOO
ww.google.com"
 like Gecko) Chr
```

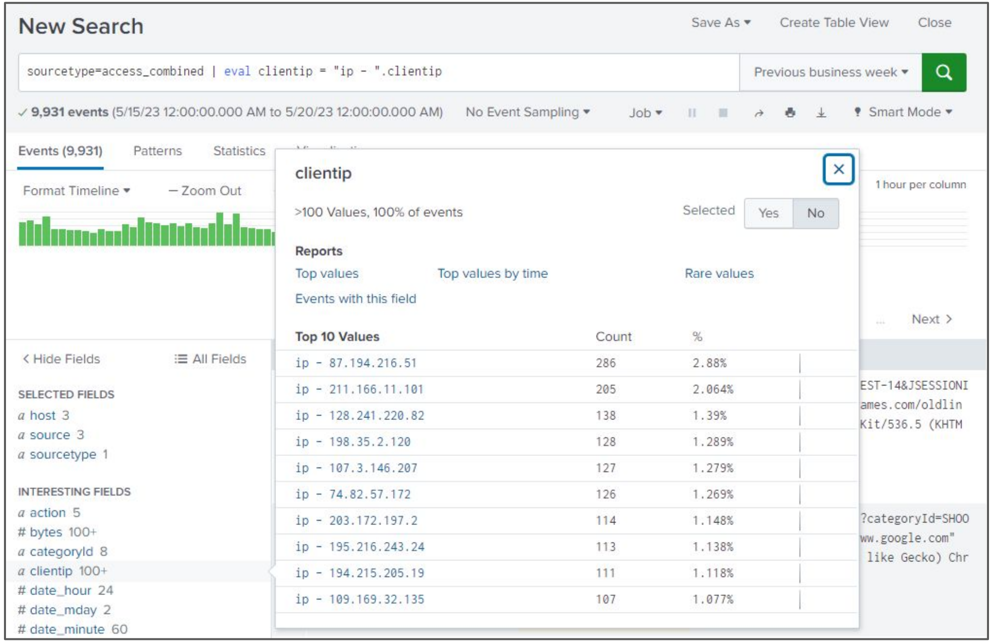
Note: The original clientip field still exists containing the original values.

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The eval command

- If the **field name** that you specify **matches** a field name that **already exists** in the search results, the **results** of the **eval expression overwrite** the values in that field. With that, the **index does not change**.
- The new values in the clientip field are only valid for the duration of this search.



```
sourcetype=access_combined | eval clientip = "ip - ".clientip
```

✓ **9,931 events** (5/15/23 12:00:00.000 AM to 5/20/23 12:00:00.000 AM)

clientip

>100 Values, 100% of events

Selected: Yes [✓ No]

Reports

Top values | Top values by time | Rare values

Events with this field

Top 10 Values	Count	%
ip - 87.194.216.51	286	2.88%
ip - 211.166.11.101	205	2.064%
ip - 128.241.220.82	138	1.39%
ip - 198.35.2.120	128	1.289%
ip - 107.3.146.207	127	1.279%
ip - 74.82.57.172	126	1.269%
ip - 203.172.197.2	114	1.148%
ip - 195.216.243.24	113	1.138%
ip - 194.215.205.19	111	1.118%
ip - 109.169.32.135	107	1.077%

SELECTED FIELDS
a host 3
a source 3
a sourcetype 1

INTERESTING FIELDS
a action 5
# bytes 100+
a categoryId 8
a clientip 100+
# date_hour 24
# date_mday 2
# date_minute 60

```
EST-14&JSESSIONI
ames.com/oldlin
Kit/536.5 (KHTM

?categoryId=SHOO
ww.google.com"
 like Gecko) Chr
```

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The eval command

- The eval command evaluates **mathematical**, **string**, and **boolean** expressions.
- It supports the following **operators**.

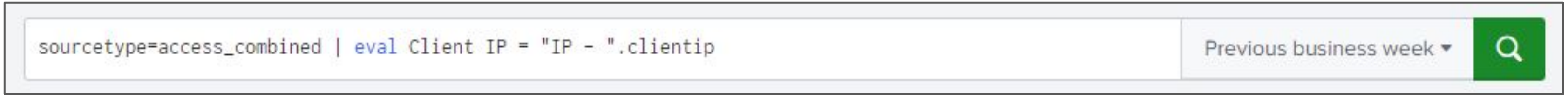
Type	Operators
Arithmetic (produce numbers)	+ - * / %
Concatenation (produce strings)	. (period) + (plus)
Boolean (produce booleans)	AND OR NOT XOR < > <= >= != = == LIKE

PER SCHOLAS

4.2 Use Basic SPL Commands (continued)

The eval command

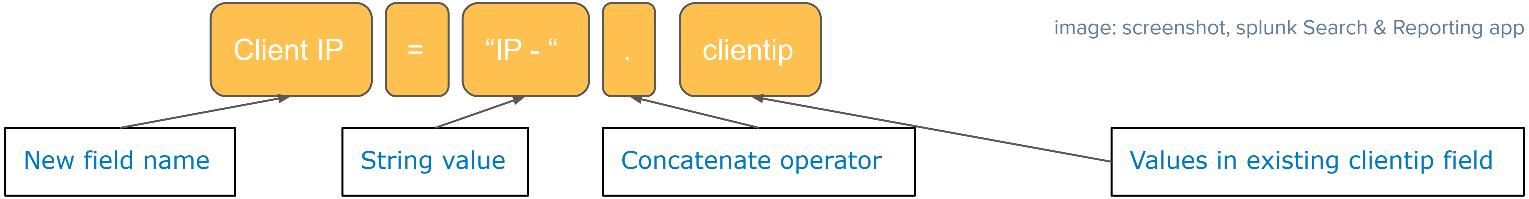
- Field values are treated in a **case-sensitive manner**
- String **values** must be **“double-quoted”**
- Field names must be **unquoted or single quotes** when they include a **special character** like a space.
- Use a **period (.)** instead of plus (+) when **concatenating** strings and numbers to avoid conflicts.
- Expressions can include values from other fields.



```
sourcetype=access_combined | eval Client IP = "IP - ".clientip
```

Previous business week ▾

image: screenshot, splunk Search & Reporting app



Client IP = “IP - “ . clientip

New field name → Client IP

String value → “IP - “

Concatenate operator → .

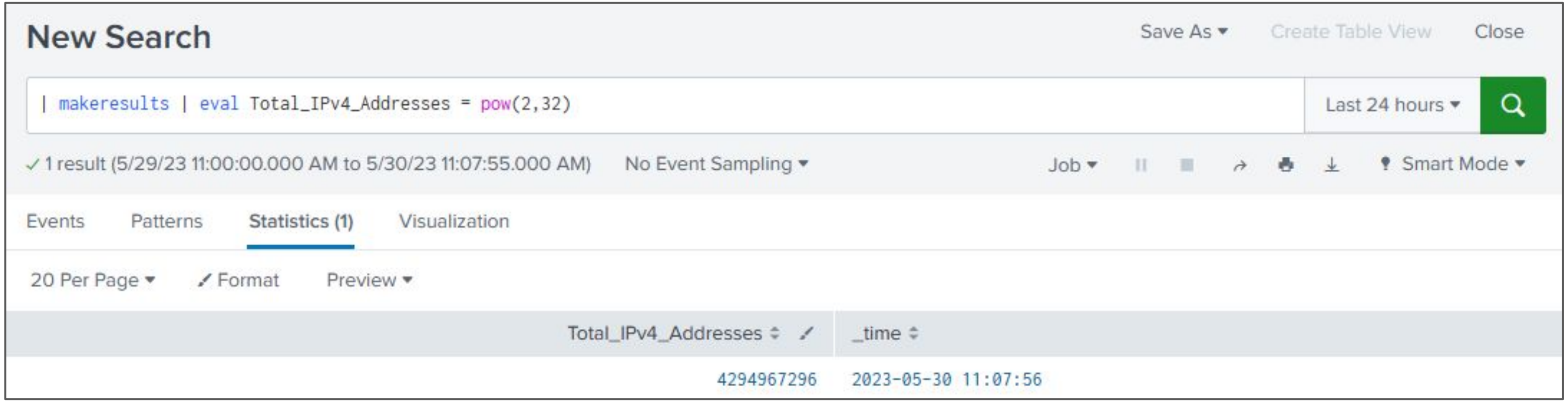
Values in existing clientip field → clientip

PER SCHOLAS

4.2 Use Basic SPL Commands (continued)

The eval command - examples.

- Eval with the **pow(<num>,<exp>)** function - Returns <num> to the power of <exp>.



```
| makeresults | eval Total_IPv4_Addresses = pow(2,32)
```

✓ 1 result (5/29/23 11:00:00.000 AM to 5/30/23 11:07:55.000 AM)

Total_IPv4_Addresses	_time
4294967296	2023-05-30 11:07:56

Note: The makeresults command (in this case) runs on the local machine and generates one result with only the _time field.
It is used here for demonstration only.

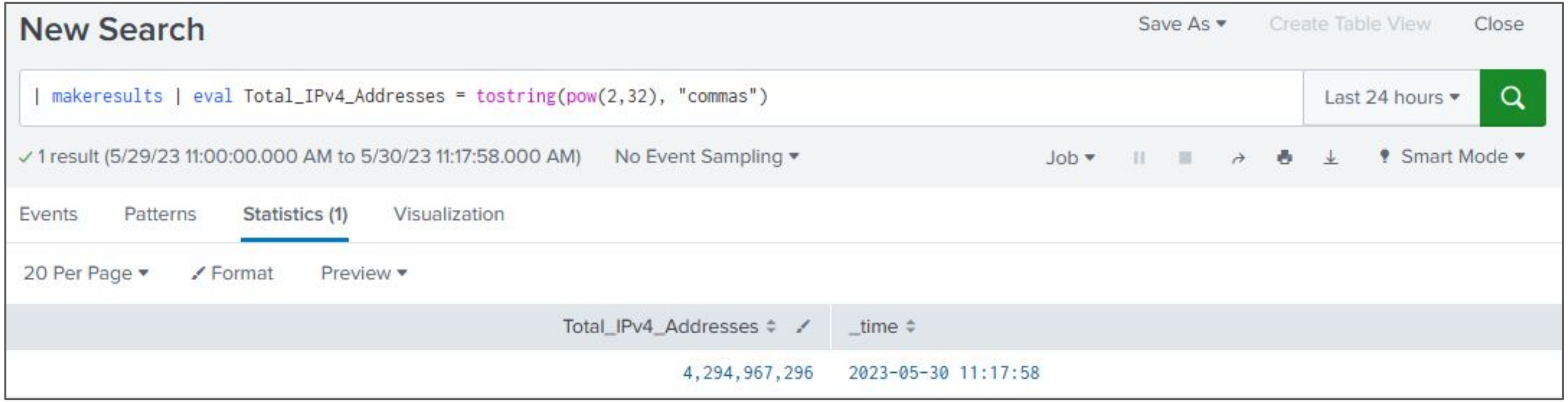
image: screenshot, splunk Search & Reporting app

PER SCHOLAS

4.2 Use Basic SPL Commands (continued)

The eval command - examples.

- Eval with the **tostring(<value>,<format>)** function - **Converts** the input <value>, such as a **number** or a **Boolean** value, to a **string**. <format> supports formatting options.



```
| makeresults | eval Total_IPv4_Addresses = tostring(pow(2,32), "commas")
```

✓ 1 result (5/29/23 11:00:00.000 AM to 5/30/23 11:17:58.000 AM)

Total_IPv4_Addresses	_time
4,294,967,296	2023-05-30 11:17:58

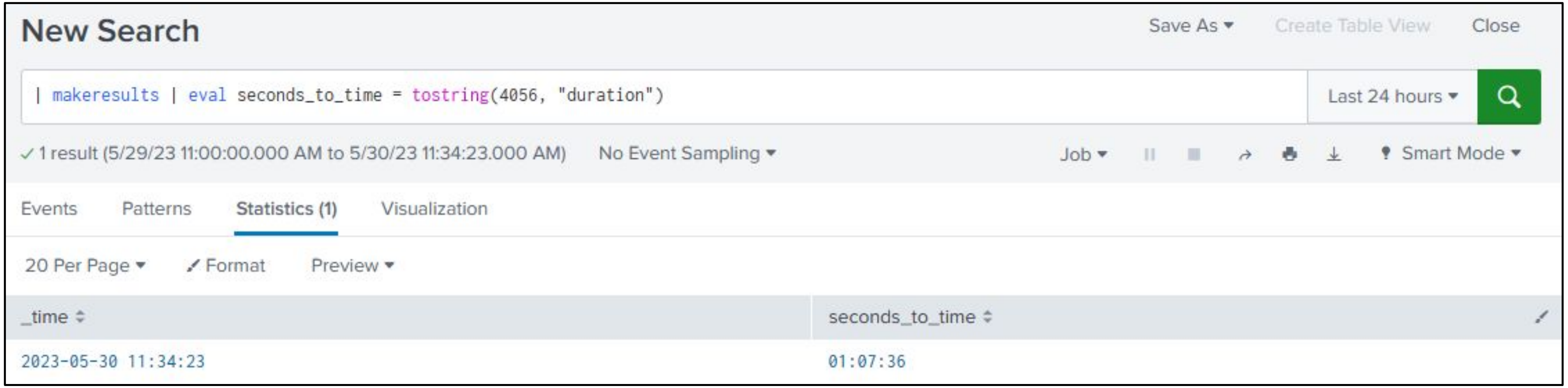
Note: Results of one function can be passed as arguments to another function. The “commas” argument will format the number by inserting commas.

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The eval command - examples.

- Eval with the **tostring(<value>,<format>)** function - **Converts** the input <value>, such as a **number** or a **Boolean** value, to a **string**. <format> supports formatting options.



```
| makeresults | eval seconds_to_time = tostring(4056, "duration")
```

✓ 1 result (5/29/23 11:00:00.000 AM to 5/30/23 11:34:23.000 AM)

_time	seconds_to_time
2023-05-30 11:34:23	01:07:36

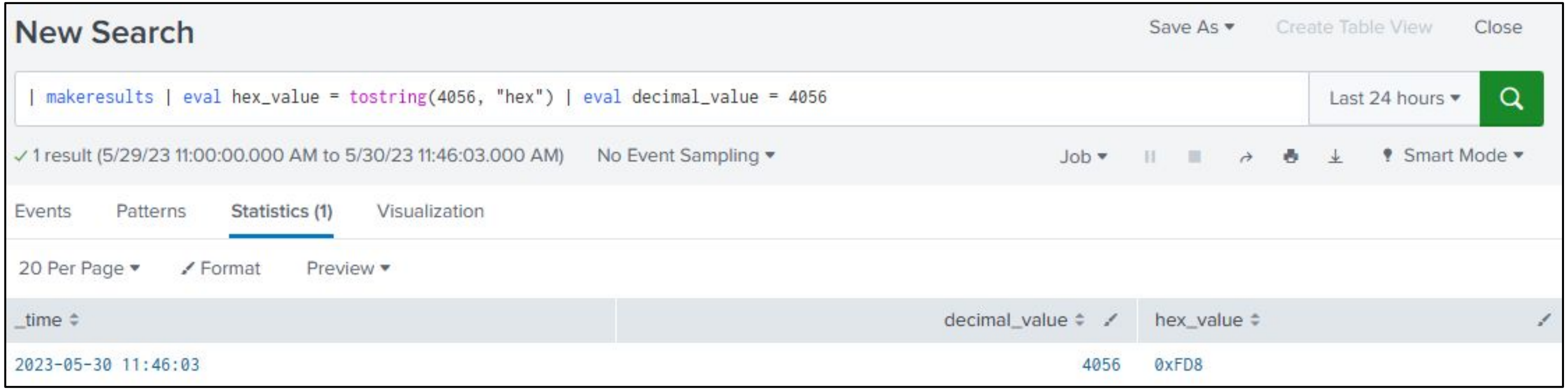
Note: the “duration” option will convert seconds to HH:MM:SS format

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The eval command - examples.

- Eval with the **tostring(<value>,<format>)** function - **Converts** the input <value>, such as a **number** or a **Boolean** value, to a **string**. <format> supports formatting options.



```
| makeresults | eval hex_value = tostring(4056, "hex") | eval decimal_value = 4056
```

_time	decimal_value	hex_value
2023-05-30 11:46:03	4056	0xFD8

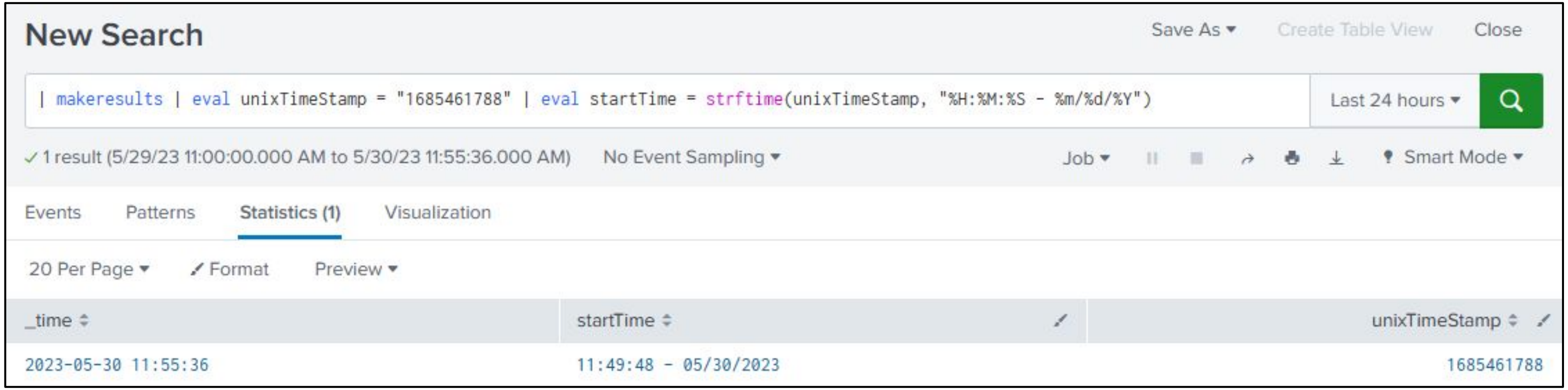
Note: The “hex” option will convert a value to Hexadecimal format

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The eval command - examples.

- Eval with the **strftime(<time>,<format>)** function - **Converts** a UNIX time value as the first **argument** and renders the **time as a string** using the **format** specified. The UNIX time must be in seconds.



```
| makeresults | eval unixTimeStamp = "1685461788" | eval startTime = strftime(unixTimeStamp, "%H:%M:%S - %m/%d/%Y")
```

✓ 1 result (5/29/23 11:00:00.000 AM to 5/30/23 11:55:36.000 AM)

_time	startTime	unixTimeStamp
2023-05-30 11:55:36	11:49:48 - 05/30/2023	1685461788

Note: For a list and descriptions of format options, see Date and time format variables.

image: screenshot, splunk Search & Reporting app

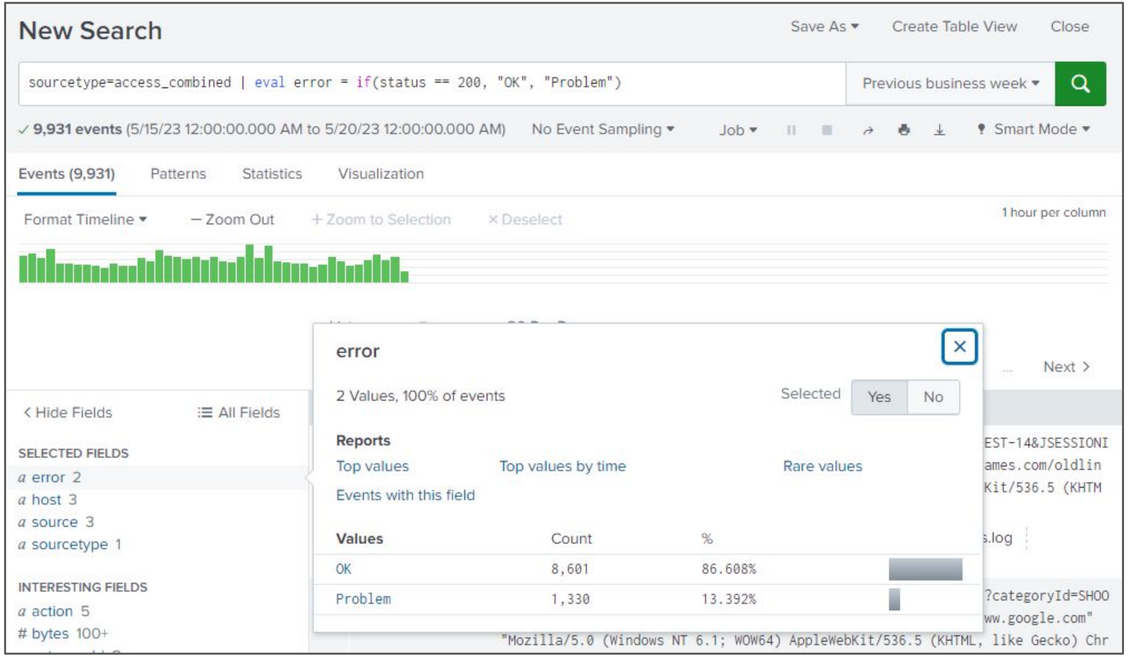
PER SCHOLAS

4.2 Use Basic SPL Commands (continued)

The eval command - examples.

Use the **if function** to analyze field values.

- Create a field called **error** in each event.
- Using the **if function**, set the **value** in the **error** field to **OK** if the **status value is 200**; otherwise, set the error field **value** to **Problem**.



New Search

```
sourcetype=access_combined | eval error = if(status == 200, "OK", "Problem")
```

Previous business week

✓ 9,931 events (5/15/23 12:00:00.000 AM to 5/20/23 12:00:00.000 AM)

error

2 Values, 100% of events

Selected: Yes No

Values	Count	%
OK	8,601	86.608%
Problem	1,330	13.392%

SELECTED FIELDS

- a error 2
- a host 3
- a source 3
- a sourcetype 1

INTERESTING FIELDS

- a action 5
- # bytes 100+

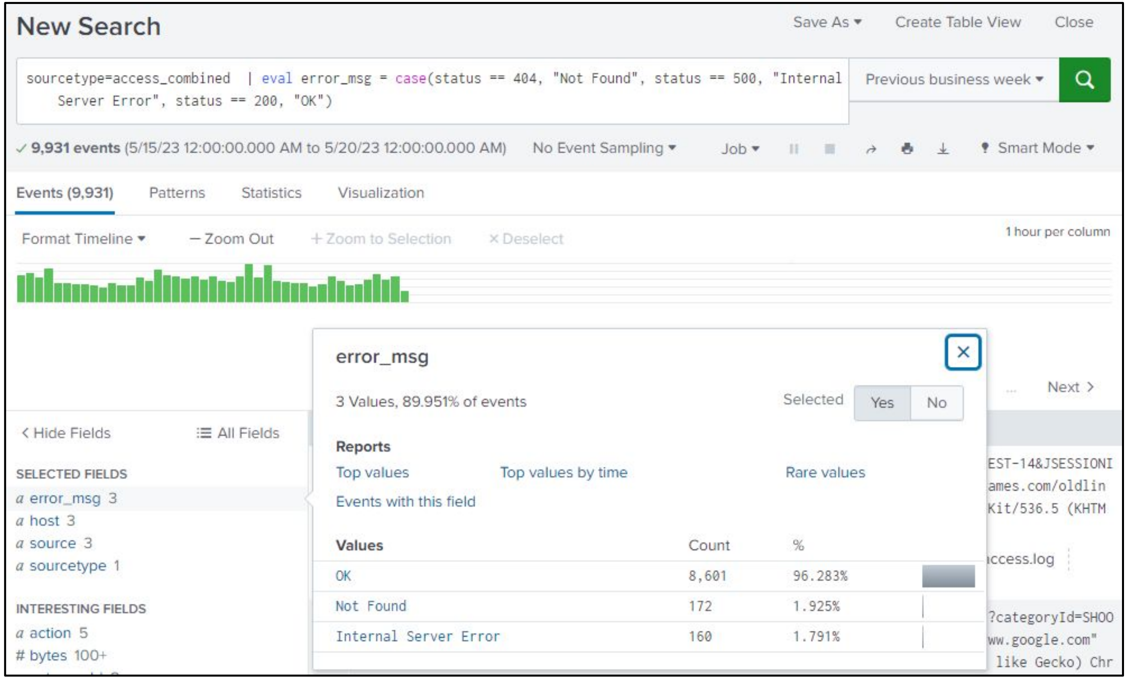
image: screenshot, splunk Search & Reporting app
The error field is selected to produce this image.

4.2 Use Basic SPL Commands (continued)

The eval command - examples.

Set a **status text** to some **simple http** error **codes.**

- Create a field called **error_msg** in each event.
- Using the **case function**, set the **value** in the **error** field to **OK** if the **status value is 200**, **Not Found** if the **value** is **404**, and **Internal Server Error** if the **value** is **500**.



```
sourcetype=access_combined  | eval error_msg = case(status == 404, "Not Found", status == 500, "Internal
    Server Error", status == 200, "OK")
```

✓ **9,931 events** (5/15/23 12:00:00.000 AM to 5/20/23 12:00:00.000 AM)

SELECTED FIELDS
a error_msg 3
a host 3
a source 3
a sourcetype 1

INTERESTING FIELDS
a action 5
bytes 100+

error_msg

3 Values, 89.951% of events

Selected [✓ Yes] No

Values	Count	%
OK	8,601	96.283%
Not Found	172	1.925%
Internal Server Error	160	1.791%

```
EST-14&JSESSIONI
ames.com/oldlin
Kit/536.5 (KHTM
ccess.log
?categoryId=SHOO
ww.google.com"
 like Gecko) Chr
```

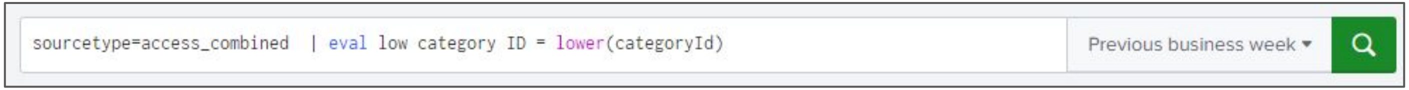
image: screenshot, splunk Search & Reporting app
The error_msg field is selected to produce this image.

4.2 Use Basic SPL Commands (continued)

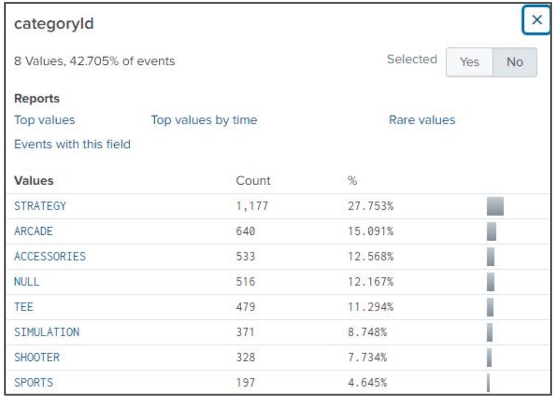
The eval command - examples.

Convert values to **lowercase**.

- Create a field called **low category ID** in each event.
- Using the **lower** function, populate the field with the **lowercase version** of the values in the categoryId field.



```
sourcetype=access_combined  | eval low category ID = lower(categoryId)
```



categoryId ✕

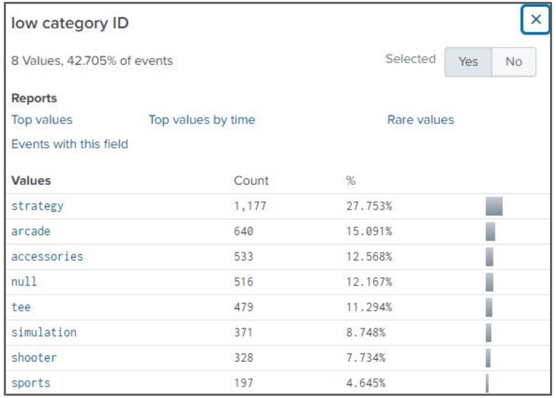
8 Values, 42.705% of events — Selected: Yes No

Reports

Top values — Top values by time — Rare values

Events with this field

Values	Count	%
STRATEGY	1,177	27.753%
ARCADE	640	15.091%
ACCESSORIES	533	12.568%
NULL	516	12.167%
TEE	479	11.294%
SIMULATION	371	8.748%
SHOOTER	328	7.734%
SPORTS	197	4.645%



low category ID ✕

8 Values, 42.705% of events — Selected: Yes No

Reports

Top values — Top values by time — Rare values

Events with this field

Values	Count	%
strategy	1,177	27.753%
arcade	640	15.091%
accessories	533	12.568%
null	516	12.167%
tee	479	11.294%
simulation	371	8.748%
shooter	328	7.734%
sports	197	4.645%

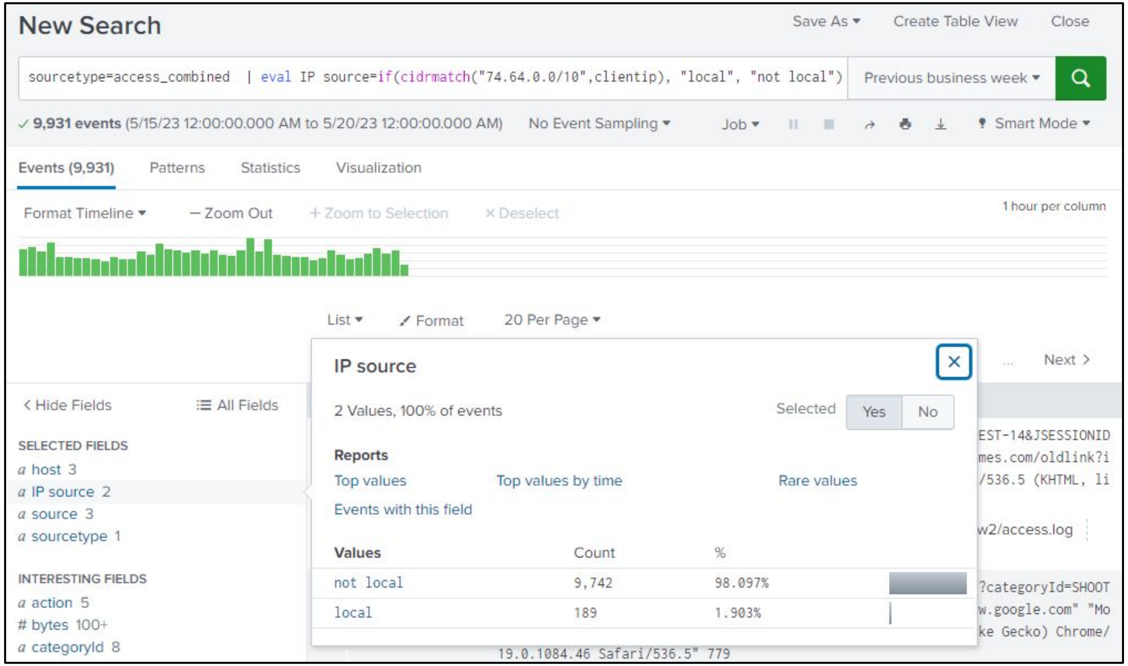
image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands (continued)

The eval command - examples.

Identify the **source** of IP addresses in the events (local/not local).

- Create an **IP source** field in the event list.
- Using the **if function** with the **cidermatch function**, any IP address in the 74.64.0.0/10 block populates the IP source field with the value "**local.**" Any other IP address will populate the field with the value "**not local.**"



New Search

```
sourcetype=access_combined  | eval IP source=if(cidrmatch("74.64.0.0/10",clientip), "local", "not local")
```

Previous business week

✓ **9,931 events** (5/15/23 12:00:00.000 AM to 5/20/23 12:00:00.000 AM)

IP source

2 Values, 100% of events

Values	Count	%
not local	9,742	98.097%
local	189	1.903%

image: screenshot, splunk Search & Reporting app

4.2 Use Basic SPL Commands - Summary

- SPL (Search Processing Language) is the search language used in Splunk for querying and analyzing data.
- The Splunk search pipeline refers to the sequential flow of data processing and transformation in a search.
- A search consists of a series of commands that are delimited by pipe (|) characters.
- The fields command is used to include or exclude specific fields in the search results.
- The dedup command removes the events that contain an identical combination of values for the fields that you specify.
- The sort command sorts all of the results by the specified fields.
- The eval command calculates an expression and puts the resulting value into a search results field.

Knowledge Check

- What character in Splunk passes the output of a section in the search as input for the next section of the search?
- What are the benefits of using the fields command?
- If you omit the [+/-] options from the fields command, how would it operate by default?
- When using the fields command, what is the delimiting character used to specify more than one field?
- What does the dedup command do?
- What are some of the options supported by the dedup command?
- Which operator is used with the eval command to concatenate strings?
- What types of expressions can the eval command evaluate?
- What are some examples of functions supported by the eval command?