Training – Day #2

15 June 2023 04:30

Note :

(1) Index name should be in lowercase. It will allow to create but "Add Data" will not push the data to splunk.

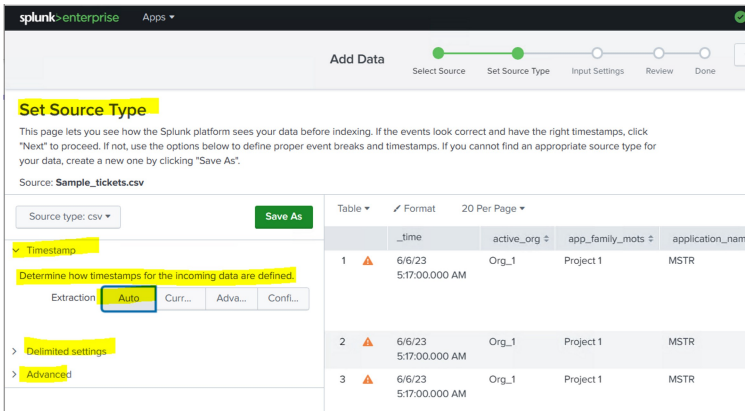
(2) If Imported data is CSV type, then first column will be considered as Header/Feld Name.

(3) ***_time*** field will be taken **from file** or **at the time when file was imported** or we can define **custom** also.

(4) Splunk has default settings to identify source type, but we can create our own source type from ***Settings => Data => Source types***

(5) Source-type details are mentioned in ***props.conf***

(6) While importing data, we can set Time for imported data.



Download Data from => https://1drv.ms/f/s!AuaadykgoQfEry6ikeBpOmerib7w?e=uOaI9U

Difference between Real-Time & Relative-Time search :

Real-time search => Search at the time of insertion, hence CPU utilization will be very high, so mostly we disable. Use real-time maximum for 1min to 2 min for security purpose.

Relative time search => Search is happening **after storage**.

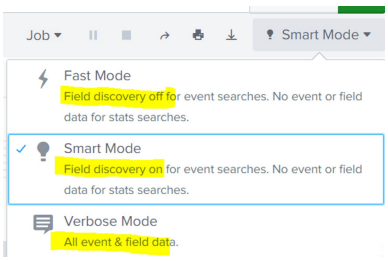
 # Search Modes:

3 types of Search mode:

(1) **Fast Mode :** Searching speed will be very less as it **skip the extraction** of filed.

(2) **Smart Mode :** Searching speed will be higher as it will **perform extraction** of filed. But I **cannot see data** for Even tab if used stats function and vice-versa

(3) **Verbose Mode :** Searching speed will be very high as it will **perform extraction** of filed. But I **can switch** between Event/Pattern/Statistics pages



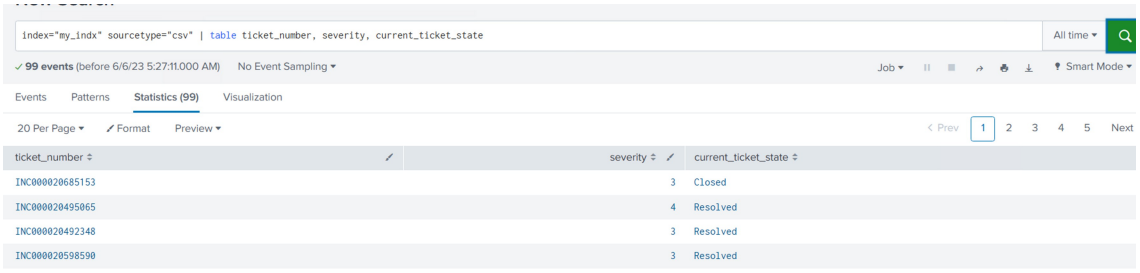
#table command:

| table <field1> <field2>

=> it will show data in tabular format for filed1 and filed2.

=> field name ***is case-sensitive.***

=> if field **name is wrong/case is wrong**, ***it will not give error. It will show the filed name with blank data***.

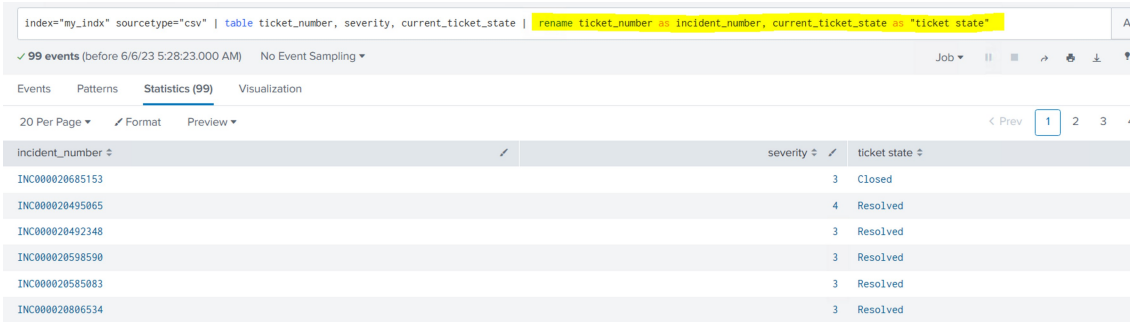


#rename command:

Rename is only for search time, not permanent.

Events and Statistics tab will show renamed fields.

| rename <field1> as name_1, <field2> as "name 1"



#stats command:

Stats command will give statistical output. It is having 5 features :

(1) ***count***

(2) ***sum***

(3) ***avg***

(4) ***list*** => list the fields

(5) ***values*** => provide Unique values

(1) ***count***

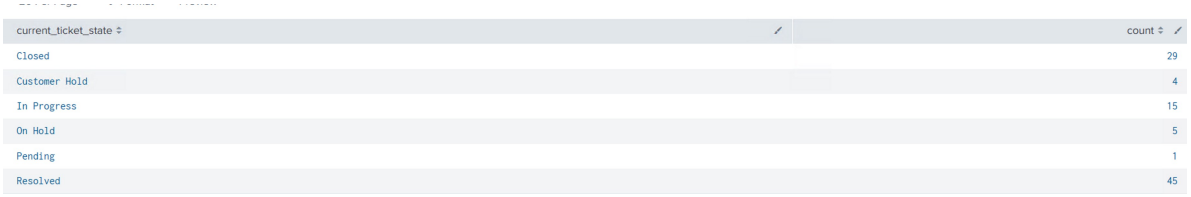
index="my_indx" sourcetype="csv" | stats count | rename count as Total

OR

index="my_indx" sourcetype="csv" | stats count as Total

by clause: will show count by grouping the fileds.

index="my_indx" sourcetype="csv" | stats count by current_ticket_state



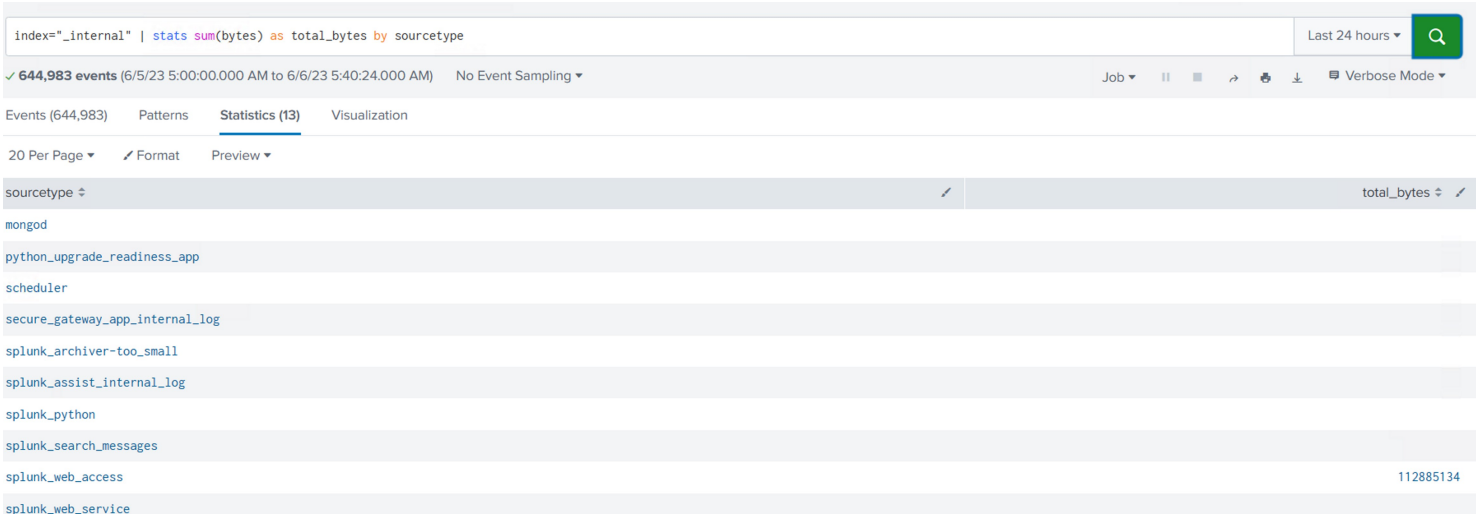
index="my_indx" sourcetype="csv" | stats count by current_ticket_state, severity

current_ticket_state	severity	count
Closed	2	7
Closed	3	16
Closed	4	6
Customer Hold	3	2
Customer Hold	4	2
In Progress	1	1

(2, 3) ***sum() and avg()***

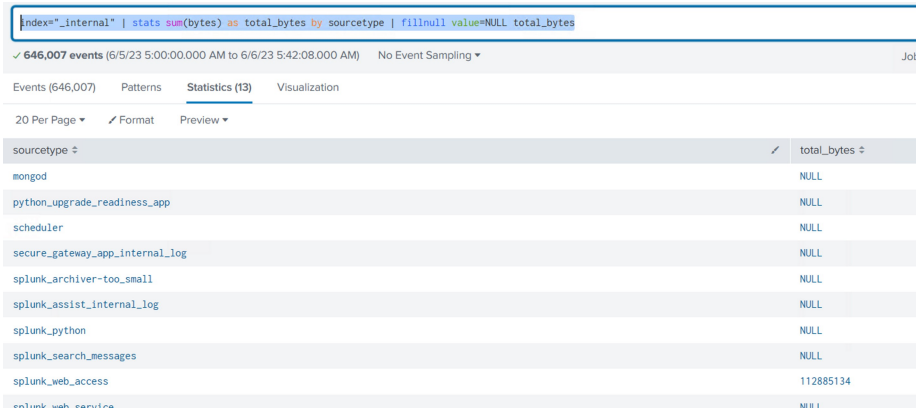
for "sum" and "avg", we need numeric filed.

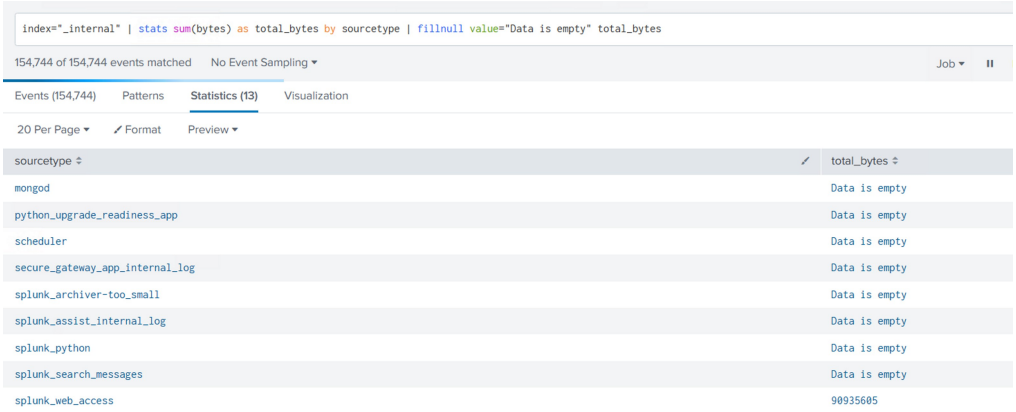
index="_internal" | stats sum(bytes) as total_bytes by sourcetype



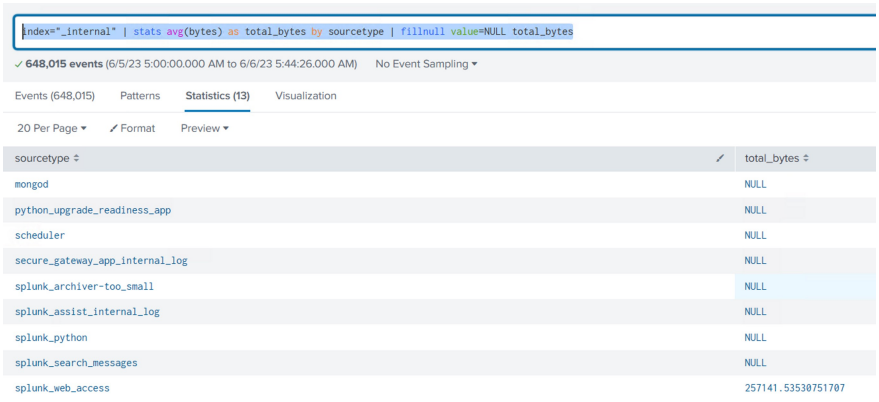
fillnull clause: will fill the null/blank values with "NULL"

index="_internal" | stats sum(bytes) as total_bytes by sourcetype | fillnull value=NULL total_bytes





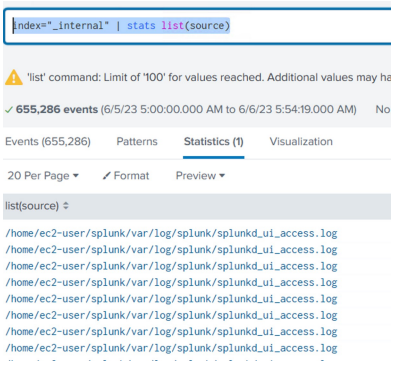
index="_internal" | stats avg(bytes) as total_bytes by sourcetype | fillnull value=NULL total_bytes



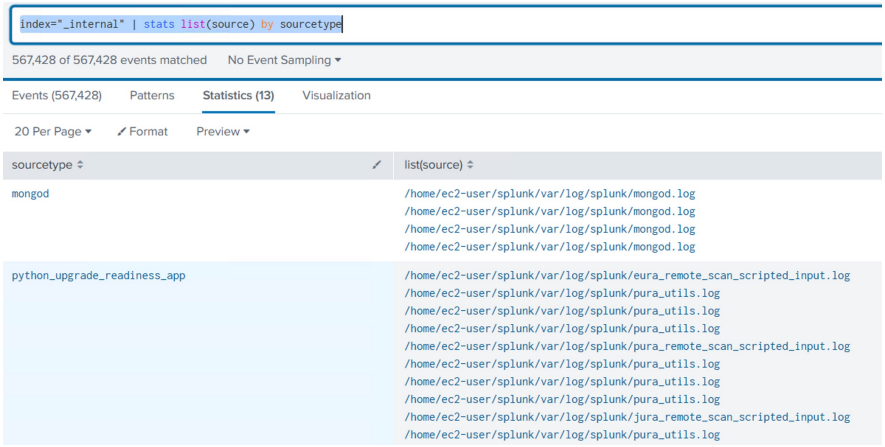
(4) *list()*

It will list all values. Values may be duplicate.

index="_internal" | stats list(source)



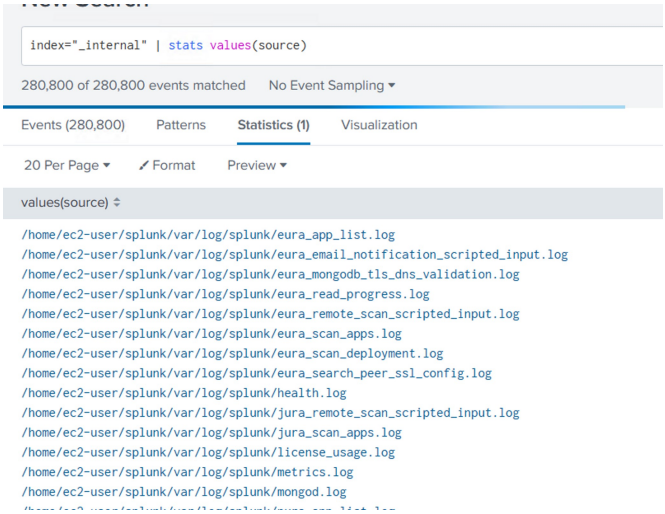
index="_internal" | stats list(source) by sourcetype



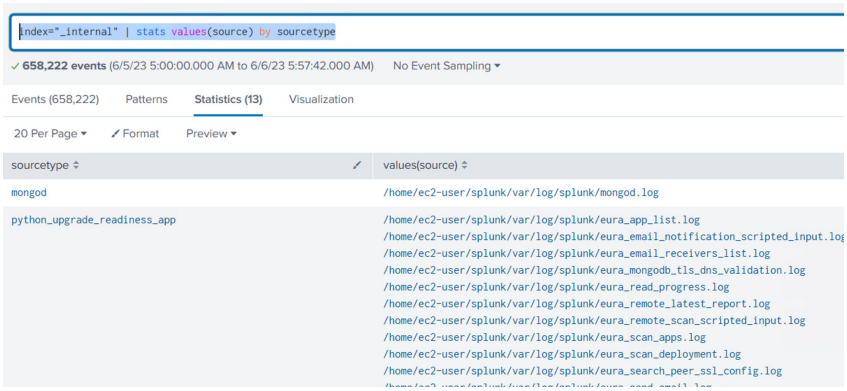
(5) ***values()***

It will list Unique values.

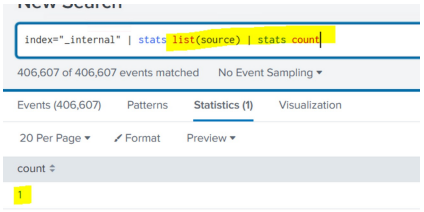
index="_internal" | stats values(source)



index="_internal" | stats values(source) by sourcetype



NOTE : dedup will not work with list. Dedup will work with normal table.



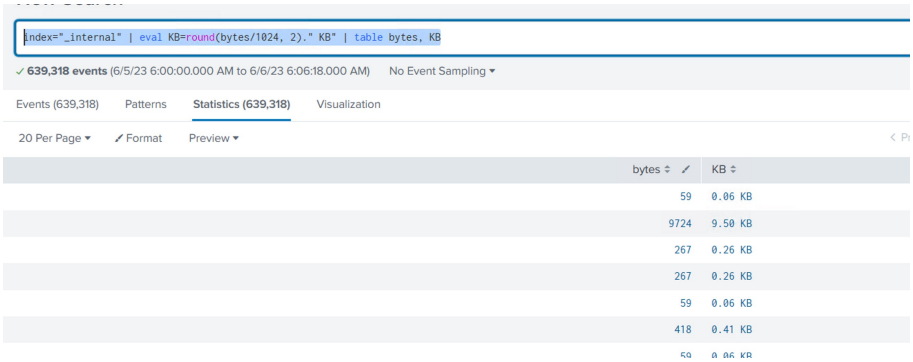
#eval command:

Eval command is ***used to initialize some variable*** and use that variable.

index="_internal" | eval KB=bytes/1024 | table bytes, KB

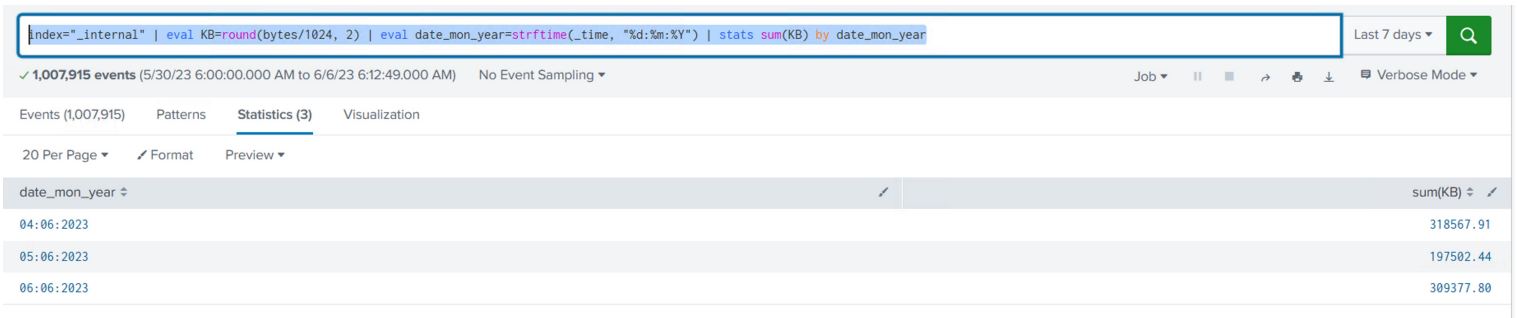
To round to 2 decimal and concatenate/ add word

index="_internal" | eval KB=round(bytes/1024, 2)." KB" | table bytes, KB



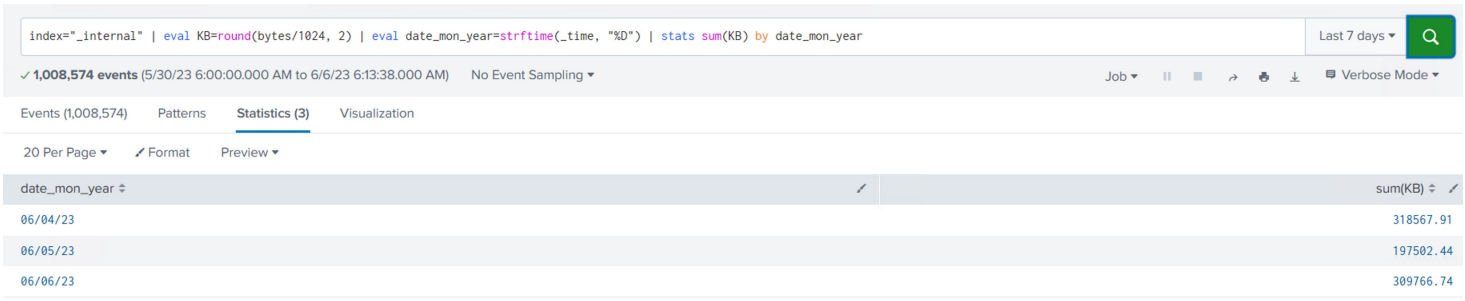
Date-Time function :
https://docs.splunk.com/Documentation/SCS/current/SearchReference/DateandTimeFunctions

index="_internal" | eval KB=round(bytes/1024, 2) | eval date_mon_year=strftime(_time, "%d:%m:%Y") | stats sum(KB) by date_mon_year



OR

index="_internal" | eval KB=round(bytes/1024, 2) | eval date_mon_year=strftime(_time, "%D") | stats sum(KB) by date_mon_year



#eval with if-else

If going to use "dedup", use it as earliest as possible.

index="my_indx" sourcetype="csv" | dedup current_ticket_state | eval state=if(current_ticket_state="Closed" OR current_ticket_state="Resolved", "Completed Ticket", "Incomplete Ticket") | table current_ticket_state, state

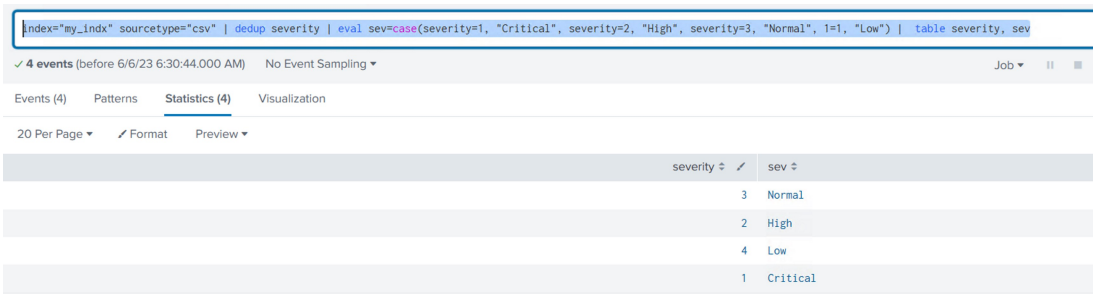
#eval with case

In "case" statement, we can define multiple values.

Case(condition1,"<value1>,condition2,"<value2>"....1=1,"<default value">

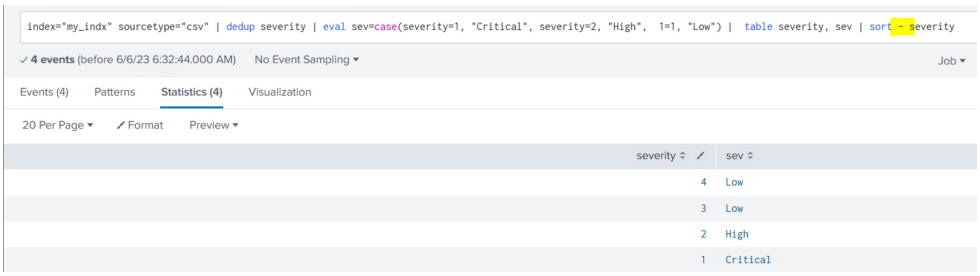
Case statement works from left to right=, hence 1=1 i.e. universal condition comes in last.

index="my_indx" sourcetype="csv" | dedup severity | eval sev=case(severity=1, "Critical", severity=2, "High", severity=3, "Normal", 1=1, "Low") | table severity, sev



Sort +/-

In sort command + is default.

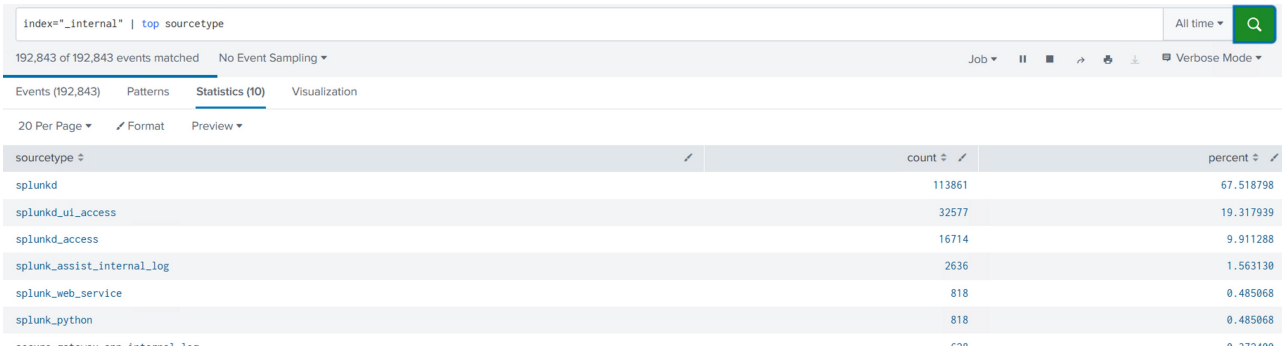


Top & Rare command

Top will give the top value.

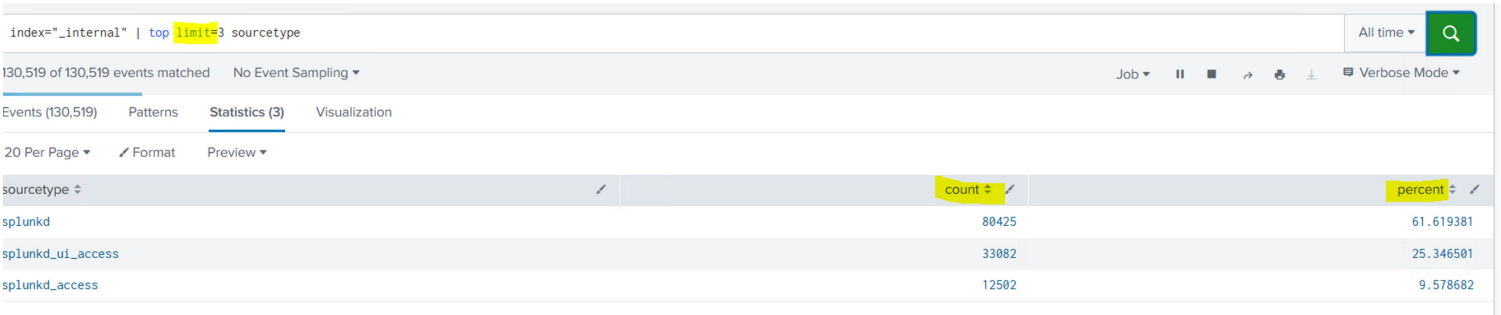
Default value of top is 10.

It will show 2 columns => count and percent columns

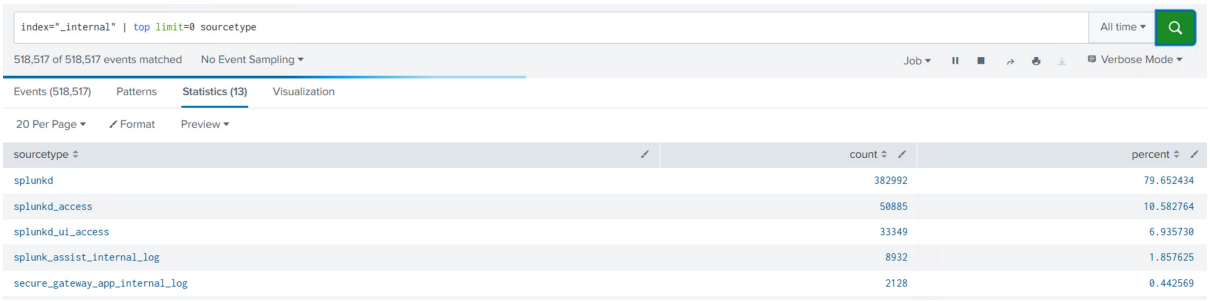


Use limit to get top n values.

index="_internal" | top limit=3 sourcetype

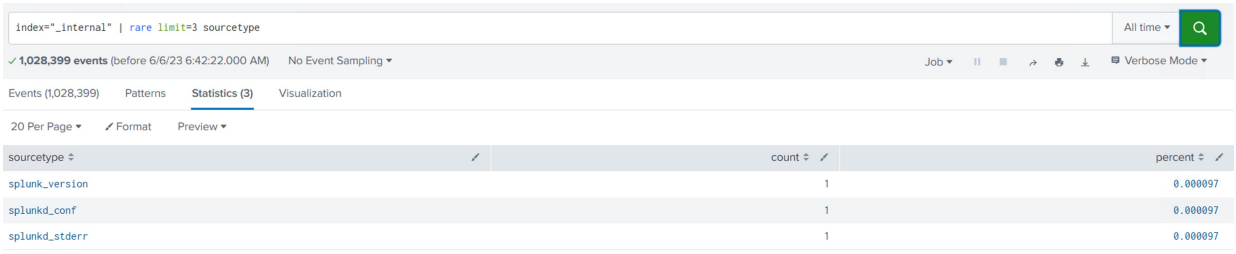


Use limit=0, to get all values.



Rare will give the bottom value

index="_internal" | rare limit=3 sourcetype

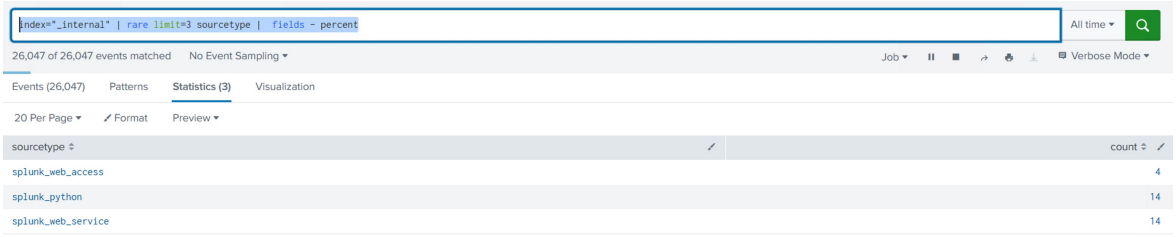


Fields +/- command

+ is to add filed. Default

- Is to remove filed

index="_internal" | rare limit=3 sourcetype | fields - percent



index="_internal" | rare limit=3 sourcetype | fields - percent, - count

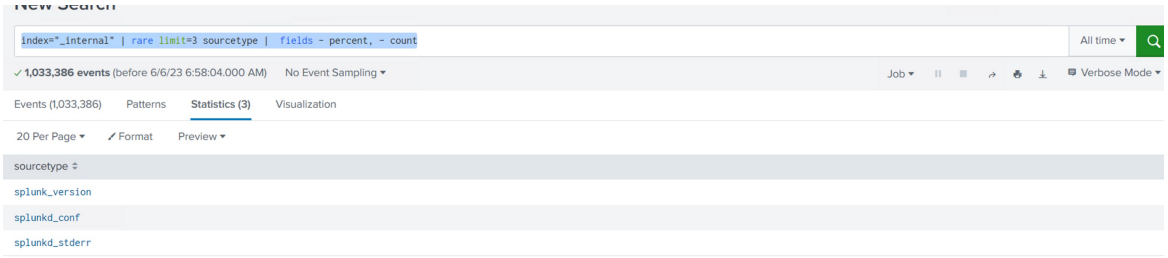


Chart command

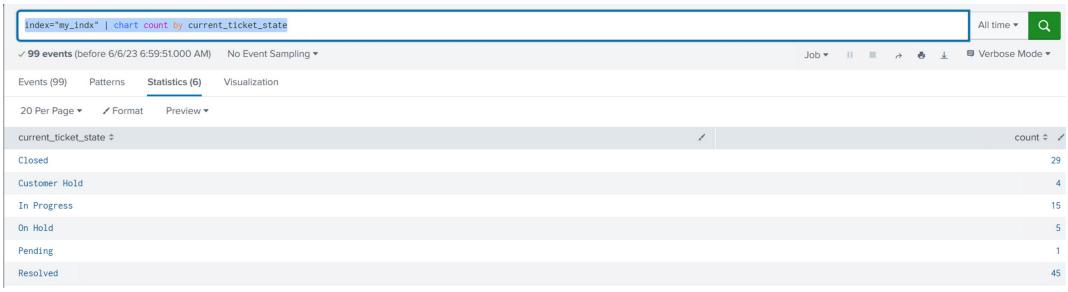
chart count by X

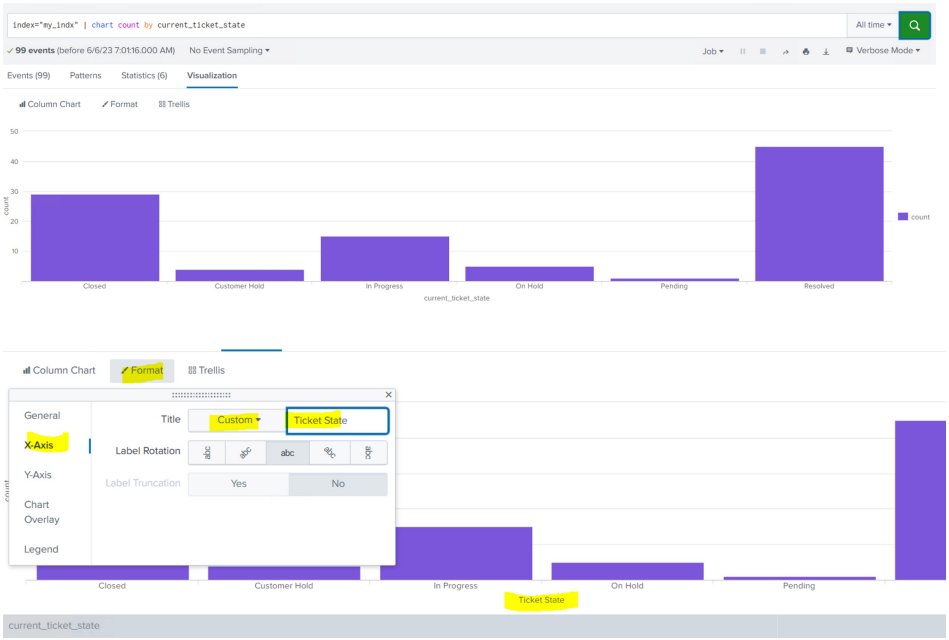
chart count(Y) by X

Count/Y will be on Y-axis

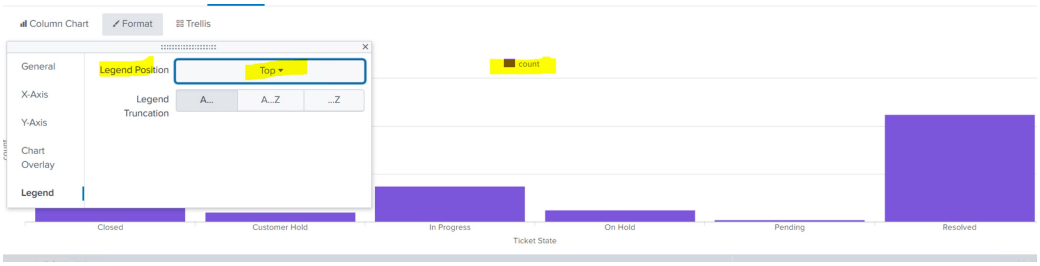
X will be on X-axis

index="my_indx" | chart count by current_ticket_state

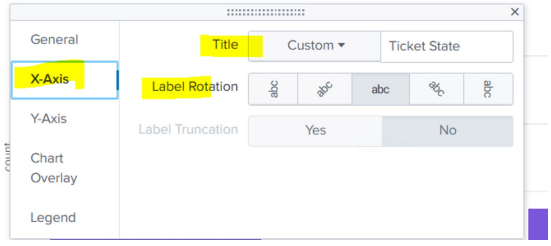




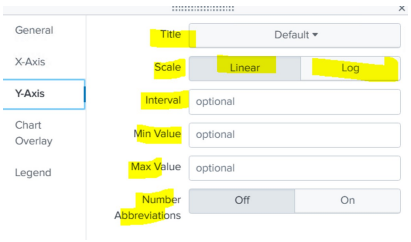
Legend Position:

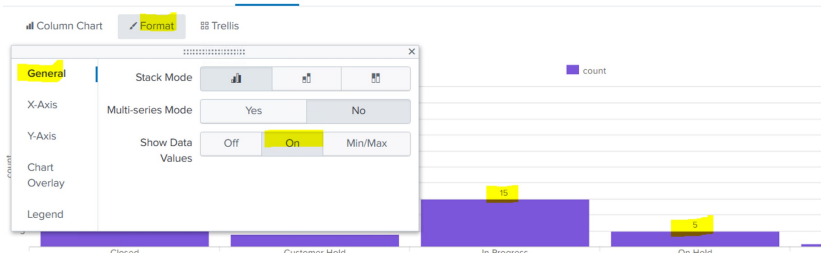


Format => X-axis Options:

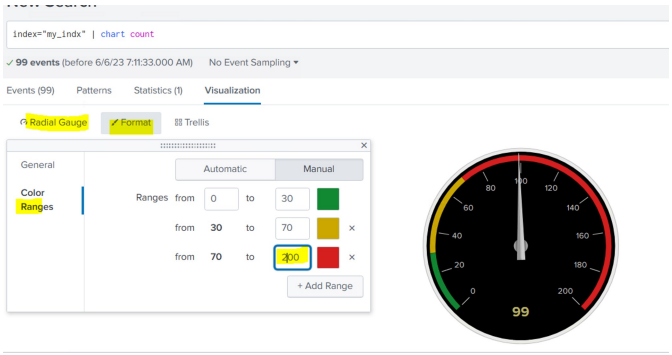


Format => Y-axis options:





Radial Gauge:



Question :

Which query is most optimized ?

SPL 1 => index="my_indx" | stats count as total by sourcetype

SPL 2 => index="my_indx" | stats count by sourcetype | rename count as total

Answer :

SPL 1 will work faster as it will generate only 2 virtual table as rename is happening in 2ns step only.
