https://aka.ms/lademo

FILTER

where is my data ?

Shows you the tables that you have within your workspace

union withsource=table *

| summarize count() by table

| sort by table asc

‘getschema’ operator

Shows you the schema that you have to work with

Syntax: [*T |] getschema*

Example: *SecurityEvent*

| getschema

‘search’ operator

Syntax: [*T |] search “string” [in (Tables)]*

Examples: *search "guest"*

SecurityEvent | where TimeGenerated >= ago(1h) | search "Guest"

search in (SecurityAlert,SecurityEvent) "guest"

‘take’ operator

Returns a random list of *n* records

Syntax: [*T |] take*

Example: *SecurityEvent | take* 10

‘top’ operator

Returns a list of the first *n* records sorted by specified column/s

Syntax: [*T |] top*

Example: *SecurityEvent | top* 10 by Account

‘where’ operator

Filters a table to the subset of rows that satisfy a predicate

Syntax: *T | where Predicate*

Examples: *SecurityEvent | where TimeGenerated > ago(1d)*

*SecurityEvent | where * contains "Victim"*

‘where’ exercise

SecurityEvent

| where TimeGenerated > ago(1d)

SecurityEvent

| where TimeGenerated > ago(1h) and EventID == 4624 // Successful logon

SecurityEvent

| where TimeGenerated > ago(1h)

| where EventID == 4624

| where AccountType =~ "user"

SecurityEvent | where EventID in (4624, 4625)

AzureNetworkAnalytics_CL | where ipv4_is_match(DestIP_s, "10.0.0.0/8")

‘extend’ operator

Create calculated columns and append them to the result set

Syntax: *T | extend ColumnName [= Expression] [, ...]*

Example: *SecurityEvent | extend ComputerNameLength = strlen(Computer)*

Perf

| where CounterName == "Free Megabytes"

| where InstanceName == "C:"

| extend FreeKB = CounterValue * 1000

| extend FreeGB = CounterValue / 1000

Perf

| where ObjectName == "LogicalDisk" and InstanceName matches regex "[A-Z]:"| project InstanceName, CounterName| where strlen(InstanceName) > 3

Perf

| where ObjectName == "LogicalDisk" and InstanceName matches regex " [A-Z]: "

| project Computer, CounterName, extract(" [A-Z]:",0,InstanceName)| take 100

ANALYZE

‘summarize’ command

Syntax: *T | summarize Aggregation [by Group Expression]*

Examples: *SecurityEvent | summarize count() by Computer*

‘summarize’ exercise

SecurityEvent

| where TimeGenerated > ago(1h)

| where EventID == 4624

| summarize dcount(Computer) by AccountType

SecurityEvent

| where TimeGenerated > ago(1h)

| where EventID == 4624

| summarize count() by AccountType, Computer

Variants and add-ons to summarize

Summarize shortcuts

SecurityEvent | project Computer, Account

SecurityEvent | distinct Computer, Account

SecurityEvent | where EventID == 4624 | count

Also useful

SecurityEvent | where EventID == 4624 | order by Computer asc

SecurityEvent | top 10 by TimeGenerated desc

‘summarize’ as a filter: arg_min(), arg_max()

Filter out top or bottom rows. Essentially “top by”.

```
SecurityEvent
| where TimeGenerated > ago(1h)
| summarize arg_max(TimeGenerated, *) by Computer
```

```
SecurityEvent
| where TimeGenerated > ago(1h)
| summarize arg_max(TimeGenerated,Computer) by IpAddress
```

‘order by’ exercise

```
SecurityAlert
| where TimeGenerated > ago(7d)
| extend severityOrder = case (
        AlertSeverity == "High", 3,
        AlertSeverity == "Medium", 2,
        AlertSeverity == "Low", 1,
        AlertSeverity == "Informational", 0,
        -1)
| order by severityOrder
| project-away severityOrder
```

‘project’ operator

Select the columns to include, rename or drop, and insert new computed columns.

Syntax: *T | project ColumnName [= Expression] [, ...]*

Example: *SecurityEvent | project TimeGenerated, Computer*

‘project’ exercise

```
SecurityEvent
| project IsImportant = iff(Account contains "Admin", true, false), Computer
```

‘summarize’ to prepare: make_list():

```
VMComputer
| summarize make_list(Ipv4Addresses,10)
```

‘summarize’ to prepare: make_set():

```
SigninLogs
| summarize make_list(IPAddress,10) by ClientAppUsed
```

Password spray detection – Example 1

```
SecurityEvent
| where TimeGenerated > ago(60d)
| where EventID == 4625
| summarize count() by TargetAccount
```

PRESENT

‘bin’ and time series

Groups values into a smaller set of specific values. It is very useful in summarize operations to create time series.

SecurityEvent

| summarize count() by bin(TimeGenerated, 1h)

| render timechart

SecurityEvent

| *summarize count()* by bin(TimeGenerated, 1h)

| *render columnchart* with (title="Security Events by the Hour")

‘bin’ exercise

SecurityEvent

| where TimeGenerated > ago(7d)

| summarize count() by bin(TimeGenerated, 1d)

VMConnection

| summarize count() by SourceIp | sort by count_ desc | render columnchart

ADVANCED OPERATIONS

‘materialize’ statement

```
let LowActivityAccounts =
 materialize(SecurityEvent
            | summarize cnt = count() by Account
    | where cnt < 10);
    LowActivityAccounts
    | where Account contains "admin"
```

‘union’ operator

Example:

```
SecurityEvent
| union (WindowsFirewall | where CommunicationDirection == "RECEIVE")
```

‘join’ operator

Syntax: LeftTable | join [JoinParameters] (RightTable) on Attributes

Example: *SecurityEvent | where TimeGenerated > ago(7d) | take 100 | join (Alert) on Computer*