Demystifying KQL for Threat Hunters

Part of the Kusto Ninja Series

Table of Contents

- Intro to Datasets
- Diving into Dynamics
- Regular Expressions
- Advanced Summaries
- Advanced Time Filters
- Time Series Analysis
- Functions

Datasets Used in this training

This training uses the Microsoft Defender XDR Advanced Hunting tables for its examples.
Please use this training in your own environment as it can query against your personal dataset.

Product Name	Function
Defender for Endpoint (MDE)	XDR and Antivirus
Defender for Identity (MDI)	On-Prem Identity Protection
Defender for Office 365	M365 Office Apps Protection
Defender for Cloud Apps	Cloud App Protection
Defender for Cloud	Cloud Posture Protection
Defender Vulnerability Management	Vulnerability Management
Azure Active Directory Identity Protection	Risky Identity Detection
Data Loss Protection	Data Leakage Prevention
Defender for IOT	Internet of Things Protection

Reference: https://learn.microsoft.com/en-us/defender-xdr/advanced-hunting-schema-tables

Diving Into Dynamics

Dynamic Data Type

The dynamic scalar data type is special in that it can take on any value of other scalar data types, as well as arrays and property bags.

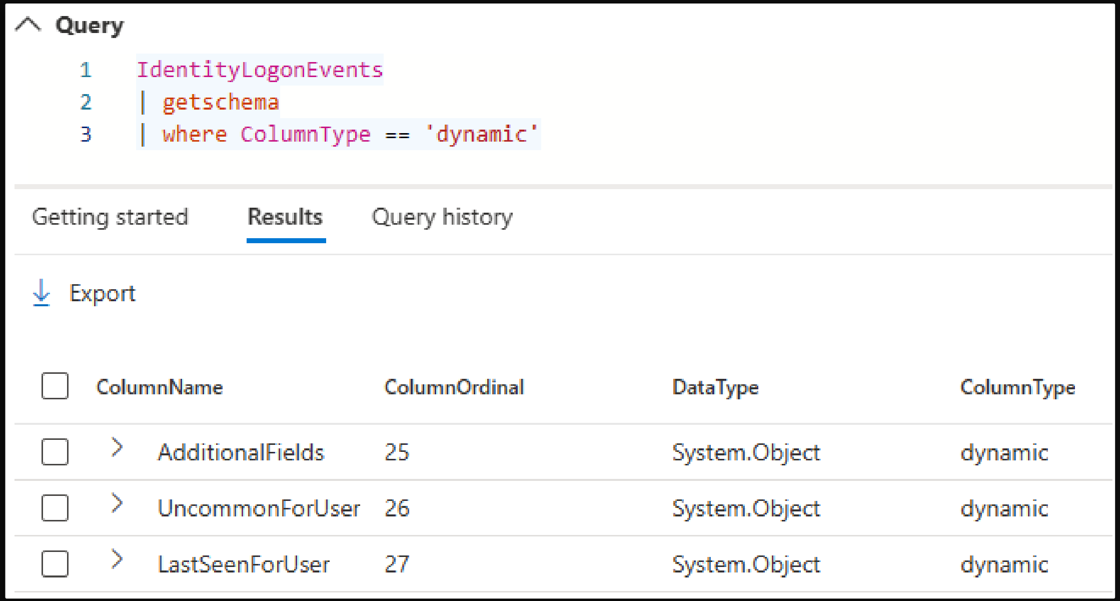
Specifically, a dynamic value can be:

•A Null (An Empty Value)

•A value of any of the primitive scalar data types: Boolean, datetime, guid, integer, long, real, string, and timespan.

•An array of dynamic values, holding zero or more values with zero-based indexing.

•A property bag that maps unique string values to dynamic values.



```
IdentityLogonEvents
| getschema
| where ColumnType == 'dynamic'
```

ColumnName	ColumnOrdinal	DataType	ColumnType
AdditionalFields	25	System.Object	dynamic
UncommonForUser	26	System.Object	dynamic
LastSeenForUser	27	System.Object	dynamic

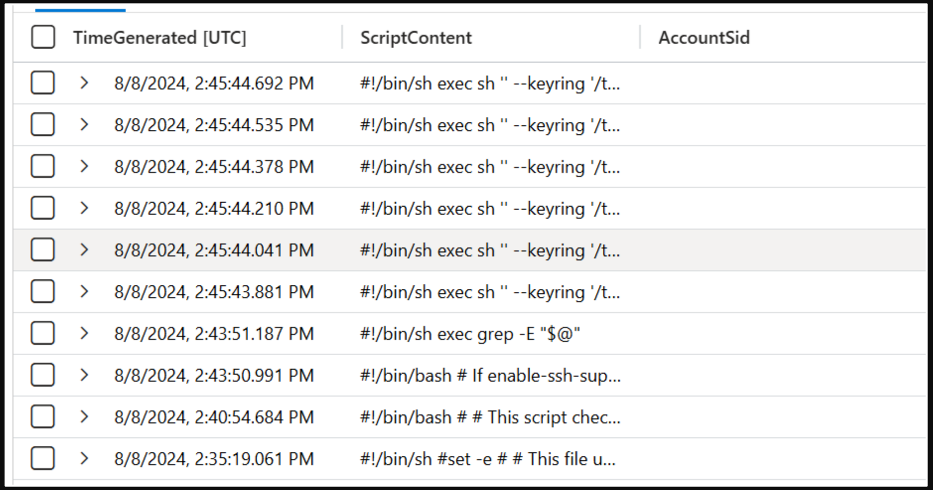
Accessing Dynamic Data

Below are the different notation types that be used to access and parse out data from dynamic fields such as json arrays.

Notation Type	Access Method	Example
Dot Notation	(Dict.key)	AdditonalFields.ScriptContent
Brackets Notation	(dict[‘key’])	(AuthenticationDetails)[0].succeeded)

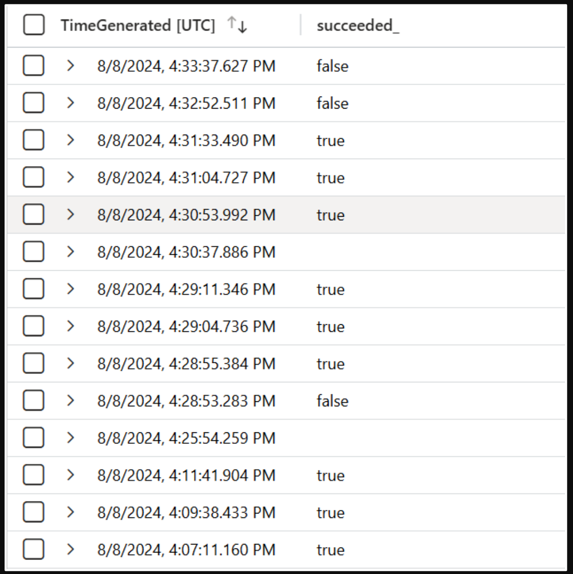
Accessing Dynamic Data Examples

```
DeviceEvents
| extend ScriptContent =
AdditionalFields['ScriptContent’]
| take 10
```



☐	TimeGenerated [UTC]	ScriptContent	AccountSid
☐	8/8/2024, 2:45:44.692 PM	#!/bin/sh exec sh '' --keyring '/t...	
☐	8/8/2024, 2:45:44.535 PM	#!/bin/sh exec sh '' --keyring '/t...	
☐	8/8/2024, 2:45:44.378 PM	#!/bin/sh exec sh '' --keyring '/t...	
☐	8/8/2024, 2:45:44.210 PM	#!/bin/sh exec sh '' --keyring '/t...	
☐	8/8/2024, 2:45:44.041 PM	#!/bin/sh exec sh '' --keyring '/t...	
☐	8/8/2024, 2:45:43.881 PM	#!/bin/sh exec sh '' --keyring '/t...	
☐	8/8/2024, 2:43:51.187 PM	#!/bin/sh exec grep -E "$@"	
☐	8/8/2024, 2:43:50.991 PM	#!/bin/bash # If enable-ssh-sup...	
☐	8/8/2024, 2:40:54.684 PM	#!/bin/bash # # This script chec...	
☐	8/8/2024, 2:35:19.061 PM	#!/bin/sh #set -e # # This file u...	

```
SigninLogs
| extend succeeded_ =
tostring(parse_json(AuthenticationDetails)
[0].succeeded)
```



☐	TimeGenerated [UTC] ↑↓	succeeded_
☐	8/8/2024, 4:33:37.627 PM	false
☐	8/8/2024, 4:32:52.511 PM	false
☐	8/8/2024, 4:31:33.490 PM	true
☐	8/8/2024, 4:31:04.727 PM	true
☐	8/8/2024, 4:30:53.992 PM	true
☐	8/8/2024, 4:30:37.886 PM	
☐	8/8/2024, 4:29:11.346 PM	true
☐	8/8/2024, 4:29:04.736 PM	true
☐	8/8/2024, 4:28:55.384 PM	true
☐	8/8/2024, 4:28:53.283 PM	false
☐	8/8/2024, 4:25:54.259 PM	
☐	8/8/2024, 4:11:41.904 PM	true
☐	8/8/2024, 4:09:38.433 PM	true
☐	8/8/2024, 4:07:11.160 PM	true

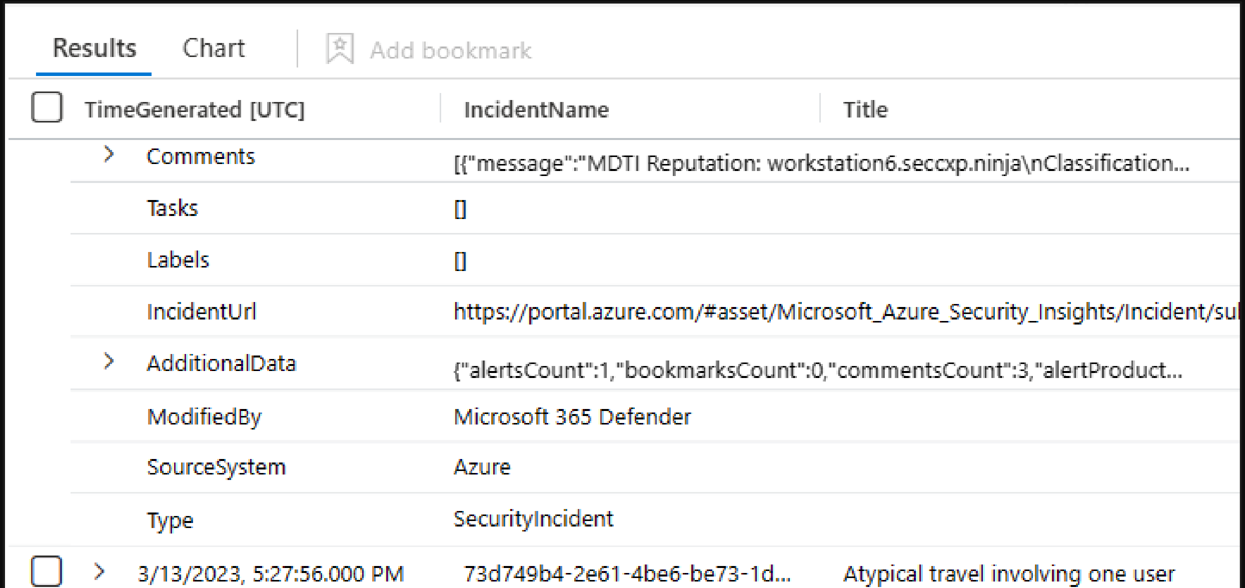
Dynamic Parsing

‘mv-expand’ operator

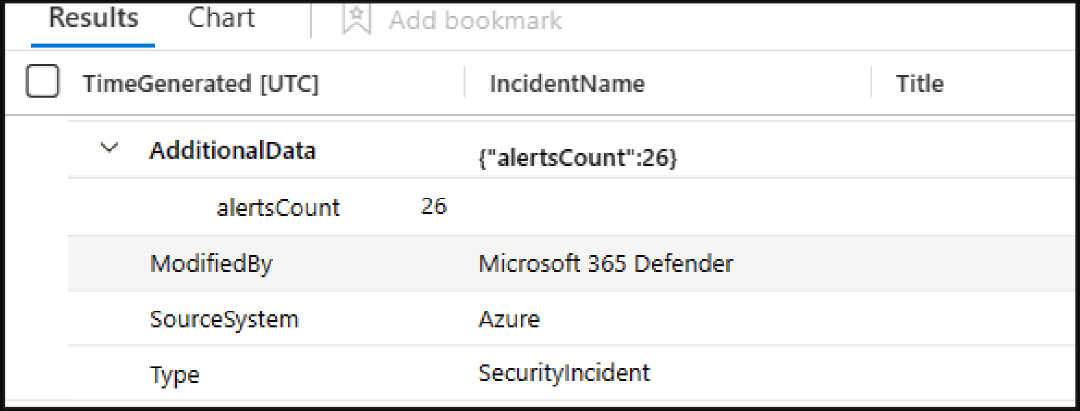
Expands multi-value dynamic arrays or property bags into multiple records.
This transforms a string into a dynamic value allows it to be used by more advanced functions.

Syntax: *Table | mv-expand entity*

Example:
SecurityIncident | mv-expand AdditionalData



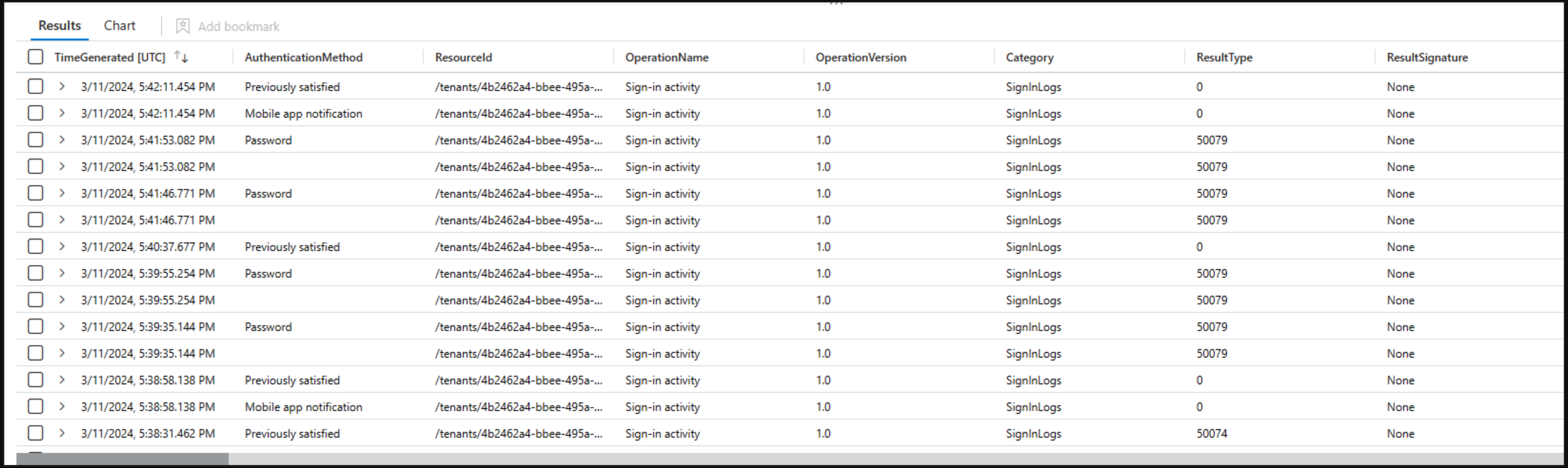
TimeGenerated [UTC]	IncidentName	Title
› Comments	[{"message":"MDTI Reputation: workstation6.seccxp.ninja\nClassification...	
Tasks	[]	
Labels	[]	
IncidentUrl	https://portal.azure.com/#asset/Microsoft_Azure_Security_Insights/Incident/su	
› AdditionalData	{"alertsCount":1,"bookmarksCount":0,"commentsCount":3,"alertProduct...	
ModifiedBy	Microsoft 365 Defender	
SourceSystem	Azure	
Type	SecurityIncident	
□ › 3/13/2023, 5:27:56.000 PM	73d749b4-2e61-4be6-be73-1d...	Atypical travel involving one user



TimeGenerated [UTC]	IncidentName	Title
⌄ AdditionalData	{"alertsCount":26}	
alertsCount	26	
ModifiedBy	Microsoft 365 Defender	
SourceSystem	Azure	
Type	SecurityIncident	

’mv-expand’ example

```
SigninLogs
| mv-expand todynamic(AuthenticationDetails)
| extend AuthenticationMethod = AuthenticationDetails
```



□		TimeGenerated [UTC] ↑↓	AuthenticationMethod	ResourceId	OperationName	OperationVersion	Category	ResultType	ResultSignature
□	›	3/11/2024, 5:42:11.454 PM	Previously satisfied	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	0	None
□	›	3/11/2024, 5:42:11.454 PM	Mobile app notification	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	0	None
□	›	3/11/2024, 5:41:53.082 PM	Password	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50079	None
□	›	3/11/2024, 5:41:53.082 PM		/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50079	None
□	›	3/11/2024, 5:41:46.771 PM	Password	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50079	None
□	›	3/11/2024, 5:41:46.771 PM		/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50079	None
□	›	3/11/2024, 5:40:37.677 PM	Previously satisfied	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	0	None
□	›	3/11/2024, 5:39:55.254 PM	Password	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50079	None
□	›	3/11/2024, 5:39:55.254 PM		/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50079	None
□	›	3/11/2024, 5:39:35.144 PM	Password	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50079	None
□	›	3/11/2024, 5:39:35.144 PM		/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50079	None
□	›	3/11/2024, 5:38:58.138 PM	Previously satisfied	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	0	None
□	›	3/11/2024, 5:38:58.138 PM	Mobile app notification	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	0	None
□	›	3/11/2024, 5:38:31.462 PM	Previously satisfied	/tenants/4b2462a4-bbee-495a-...	Sign-in activity	1.0	SignInLogs	50074	None

‘mv-apply’ operator

Applies a subquery to each record and returns the union of the results of all subqueries

Syntax:

```
Table | mv-apply [ItemIndex] ColumnsToExpand [RowLimit] on ( SubQuery )
```

Example:

```
SigninLogs
| mv-apply Location = todynamic(LocationDetails) on ( where Location.countryOrRegion == “US")
```

Reference: https://learn.microsoft.com/en-us/kusto/query/mv-expand-operator

’mv-apply’ example

```
SecurityAlert
|mv-apply entity = todynamic(Entities) on (where entity.Type == ‘account’
|extend account = strcat (entity.NTDomain, ‘\\’,  entity.Name))
```

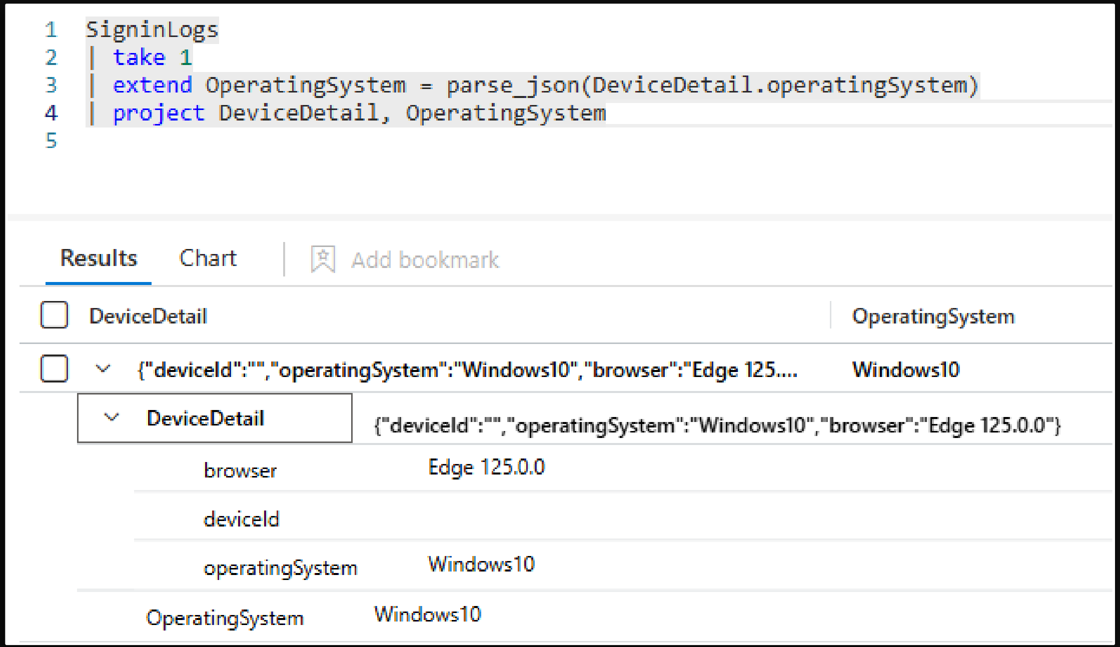
‘parse-json’ operator

Convert the string to ‘dynamic’, a value of JSON type.

This makes it easier to manipulate and create new columns.

Syntax:
Table | parse_json

Example:
SigninLogs
| extend OperatingSystem =
parse_json(DeviceDetail.operatingSystem



```
SigninLogs
| take 1
| extend OperatingSystem = parse_json(DeviceDetail.operatingSystem)
| project DeviceDetail, OperatingSystem
```

	DeviceDetail	OperatingSystem
□	{"deviceId":"","operatingSystem":"Windows10","browser":"Edge 125....	Windows10

DeviceDetail	{"deviceId":"","operatingSystem":"Windows10","browser":"Edge 125.0.0"}
browser	Edge 125.0.0
deviceId	
operatingSystem	Windows10
OperatingSystem	Windows10

’parse-json’ example

```
DeviceEvents
| where ActionType == ‘NamedPipeEvent’
| where parson_json(AdditionalFields)[‘DesiredAccess’] == 1180063
```

‘extract-json’ operator

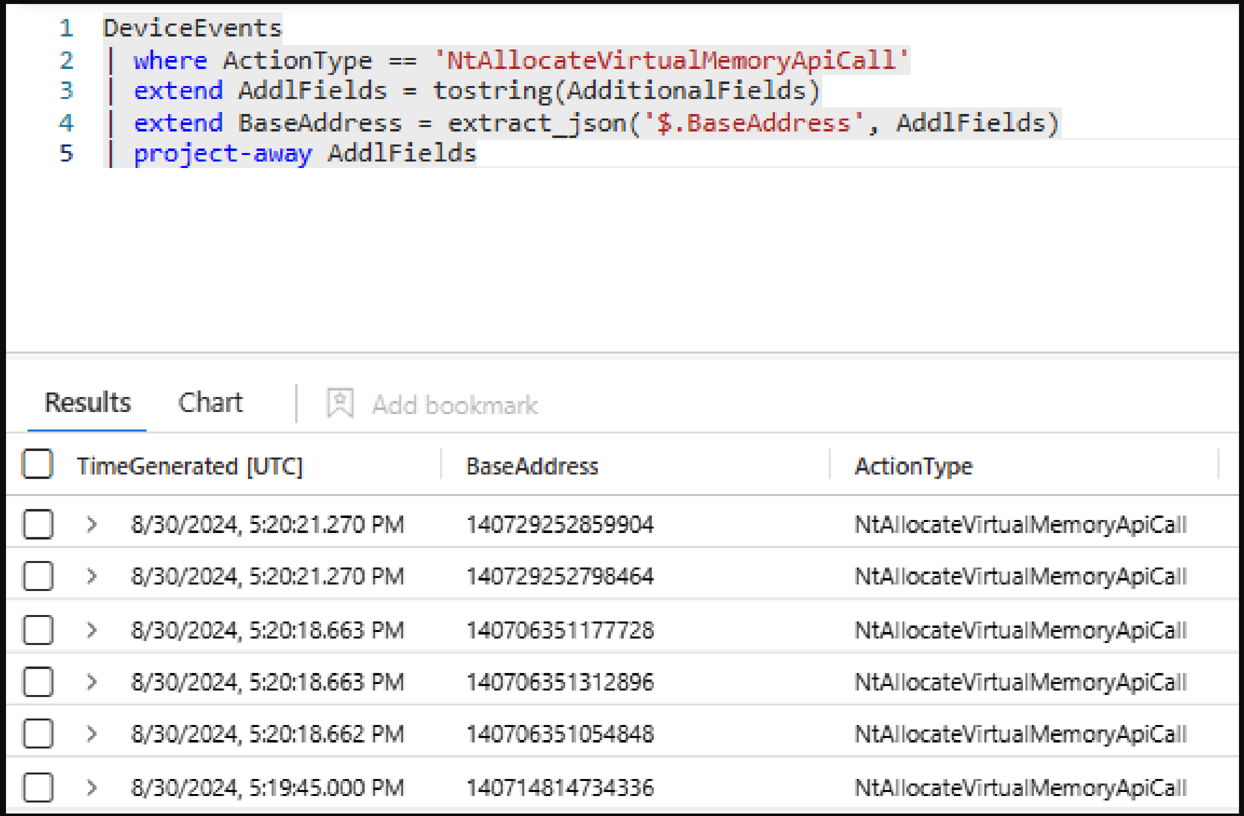
The ‘extract_json’ operator extracts a value from a JSON string.

Syntax:

Table | extract_json (jsonpath, ColumnName, typeof (DataType)

Example:

DeviceEvents
| where ActionType == 'NtAllocateVirtualMemoryApiCall'
| extend AddlFields = tostring(AdditionalFields)
| extend BaseAddress = extract_json('$.BaseAddress', AddlFields)



```
DeviceEvents
| where ActionType == 'NtAllocateVirtualMemoryApiCall'
| extend AddlFields = tostring(AdditionalFields)
| extend BaseAddress = extract_json('$.BaseAddress', AddlFields)
| project-away AddlFields
```

☐	TimeGenerated [UTC]	BaseAddress	ActionType
☐	› 8/30/2024, 5:20:21.270 PM	140729252859904	NtAllocateVirtualMemoryApiCall
☐	› 8/30/2024, 5:20:21.270 PM	140729252798464	NtAllocateVirtualMemoryApiCall
☐	› 8/30/2024, 5:20:18.663 PM	140706351177728	NtAllocateVirtualMemoryApiCall
☐	› 8/30/2024, 5:20:18.663 PM	140706351312896	NtAllocateVirtualMemoryApiCall
☐	› 8/30/2024, 5:20:18.662 PM	140706351054848	NtAllocateVirtualMemoryApiCall
☐	› 8/30/2024, 5:19:45.000 PM	140714814734336	NtAllocateVirtualMemoryApiCall

AddlFields)

’extract-json’ example

```
DeviceEvents
| where ActionType == 'NtAllocateVirtualMemoryApiCall'
| extend AddlFields = tostring(AdditionalFields)
| extend BaseAddress = extract_json('$.BaseAddress', AddlFields)
| project-away AddlFields
```

‘parse_command_line’ operator

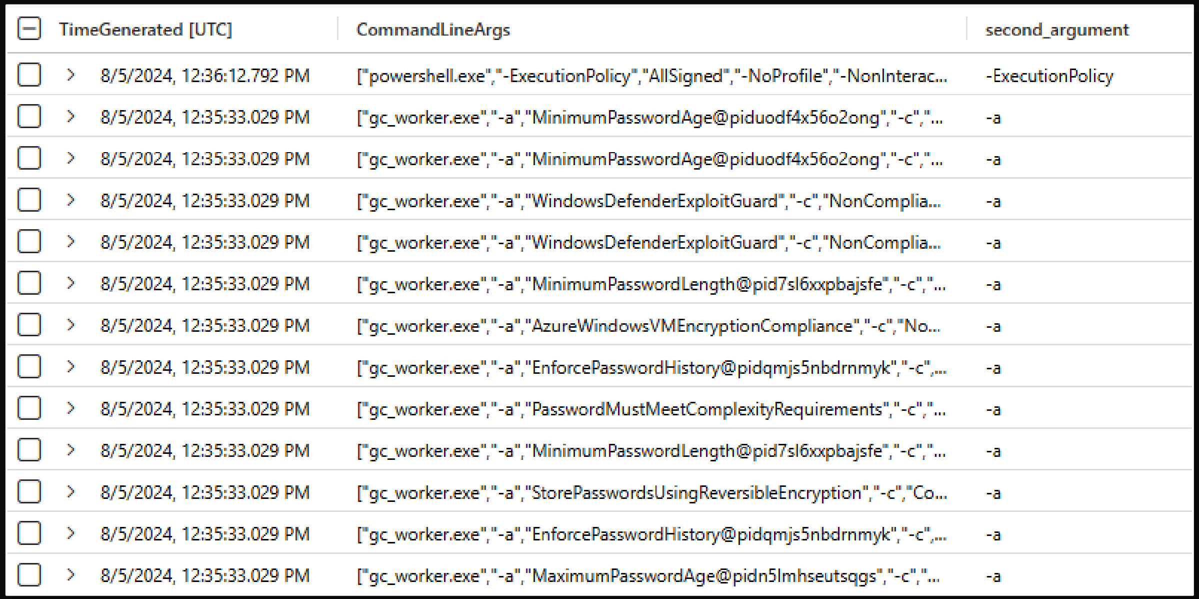
Parse a command-line string, returning the results as a dynamic array of arguments.

Syntax:

Table | parse_command_line (command_line, parser_type)

Example:

```
DeviceEvents
| where ActionType == "NtAllocateVirtualMemoryApiCall"
| extend CommandLineArgs =
parse_command_line(InitiatingProcessCommandLine,'windows')
| extend second_argument = CommandLineArgs[1]
```



⊟		TimeGenerated [UTC]	CommandLineArgs	second_argument
□	›	8/5/2024, 12:36:12.792 PM	["powershell.exe","-ExecutionPolicy","AllSigned","-NoProfile","-NonInterac...	-ExecutionPolicy
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","MinimumPasswordAge@piduodf4x56o2ong","-c","...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","MinimumPasswordAge@piduodf4x56o2ong","-c","...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","WindowsDefenderExploitGuard","-c","NonComplia...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","WindowsDefenderExploitGuard","-c","NonComplia...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","MinimumPasswordLength@pid7sl6xxpbajsfe","-c","...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","AzureWindowsVMEncryptionCompliance","-c","No...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","EnforcePasswordHistory@pidqmjs5nbdrnmyk","-c",...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","PasswordMustMeetComplexityRequirements","-c","...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","MinimumPasswordLength@pid7sl6xxpbajsfe","-c","...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","StorePasswordsUsingReversibleEncryption","-c","Co...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","EnforcePasswordHistory@pidqmjs5nbdrnmyk","-c",...	-a
□	›	8/5/2024, 12:35:33.029 PM	["gc_worker.exe","-a","MaximumPasswordAge@pidn5lmhseutsqgs","-c","...	-a

’parse_command_line’ example

```
DeviceProcessEvents
| where TimeGenerated > ago(1d) // Filter events from the last 24 hours
| extend CommandLine = parse_command_line(InitiatingProcessCommandLine,'windows')
| extend argument = CommandLine[1]
```

‘parse_path’ operator

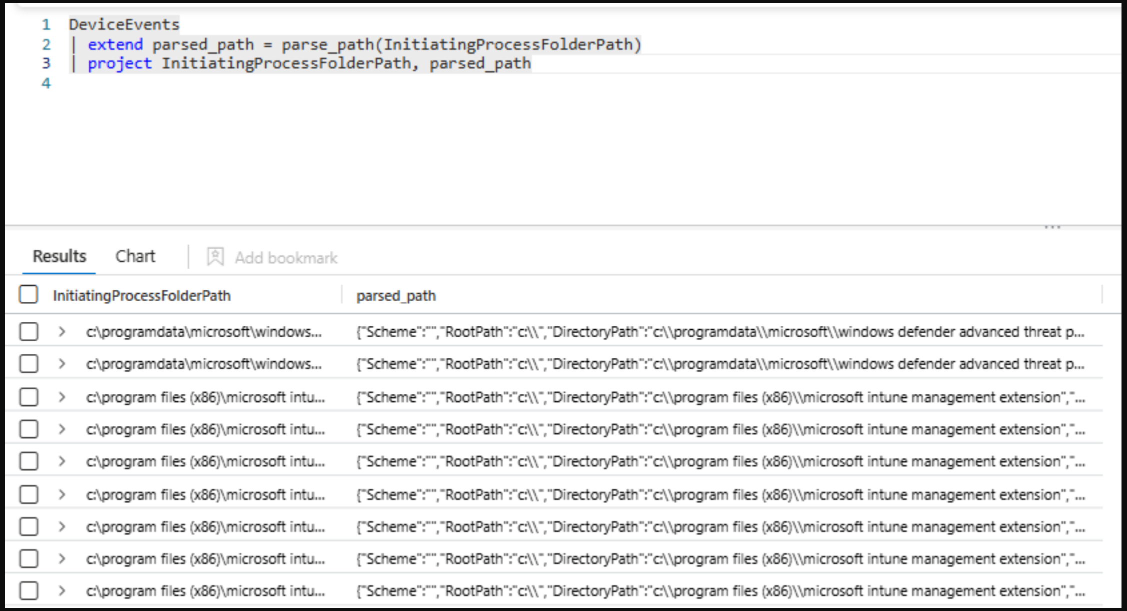
Parses a file path string and returns a dynamic object that contains parts of the path.

Syntax:

Table | parse_path(path)

Example:

```
DeviceEvents
| extend parsed_path =
parse_path(InitiatingProcessFolderPath)
```



```
DeviceEvents
| extend parsed_path = parse_path(InitiatingProcessFolderPath)
| project InitiatingProcessFolderPath, parsed_path
```

☐		InitiatingProcessFolderPath	parsed_path
☐	›	c:\programdata\microsoft\windows...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\programdata\\microsoft\\windows defender advanced threat p...
☐	›	c:\programdata\microsoft\windows...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\programdata\\microsoft\\windows defender advanced threat p...
☐	›	c:\program files (x86)\microsoft intu...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\program files (x86)\\microsoft intune management extension","...
☐	›	c:\program files (x86)\microsoft intu...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\program files (x86)\\microsoft intune management extension","...
☐	›	c:\program files (x86)\microsoft intu...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\program files (x86)\\microsoft intune management extension","...
☐	›	c:\program files (x86)\microsoft intu...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\program files (x86)\\microsoft intune management extension","...
☐	›	c:\program files (x86)\microsoft intu...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\program files (x86)\\microsoft intune management extension","...
☐	›	c:\program files (x86)\microsoft intu...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\program files (x86)\\microsoft intune management extension","...
☐	›	c:\program files (x86)\microsoft intu...	{"Scheme":"","RootPath":"c:\\","DirectoryPath":"c:\\program files (x86)\\microsoft intune management extension","...

’parse_path’ example

```
DeviceEvents
| where ActionType == "PowerShellCommand"
| extend parsed_path = parse_path(InitiatingProcessFolderPath)
| extend extension = parsed_path[ 'Extension’]
| extend file_name = parse_path(InitiatingProcessFolderPath)['Filename’]
```

Plugins

‘evaluate’ operator

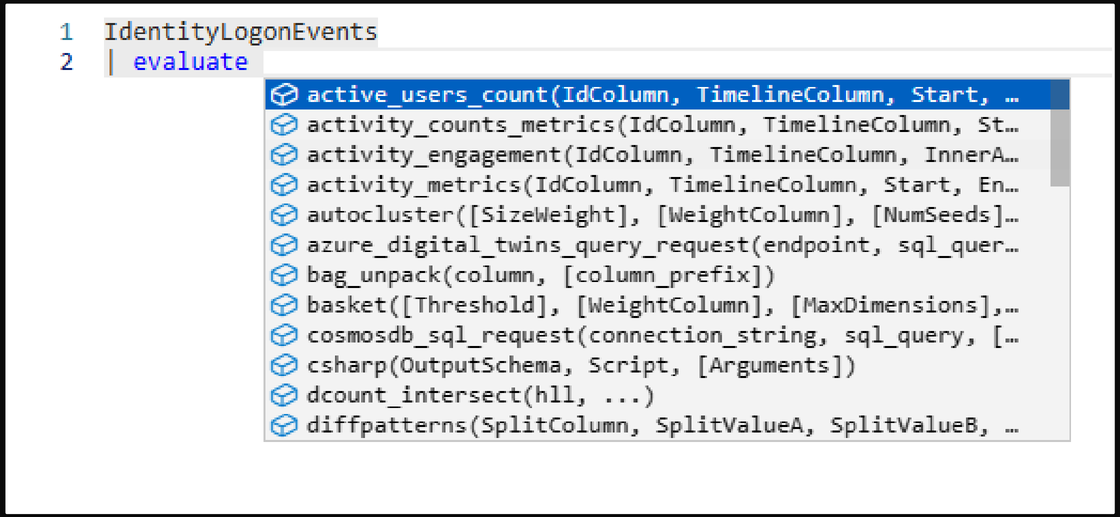
The evaluate operator is a tabular operator that allows you to invoke query language extensions known as plugins.

Syntax:

T | evaluate [evaluateParameters] PluginName ([PluginArgs])

Example:

DeviceProcessEvents | evaluate bag_unpack (AdditionalFields)



```
IdentityLogonEvents
| evaluate
```

```
active_users_count(IdColumn, TimelineColumn, Start, …
activity_counts_metrics(IdColumn, TimelineColumn, St…
activity_engagement(IdColumn, TimelineColumn, InnerA…
activity_metrics(IdColumn, TimelineColumn, Start, En…
autocluster([SizeWeight], [WeightColumn], [NumSeeds]…
azure_digital_twins_query_request(endpoint, sql_quer…
bag_unpack(column, [column_prefix])
basket([Threshold], [WeightColumn], [MaxDimensions],…
cosmosdb_sql_request(connection_string, sql_query, […
csharp(OutputSchema, Script, [Arguments])
dcount_intersect(hll, ...)
diffpatterns(SplitColumn, SplitValueA, SplitValueB, …
```

‘bag_unpack’ plugin

The ‘bag_unpack’ plugin unpacks a single column of type dynamic, by treating each property bag top-level slot as a column. The plugin is invoked with the evaluate operator:

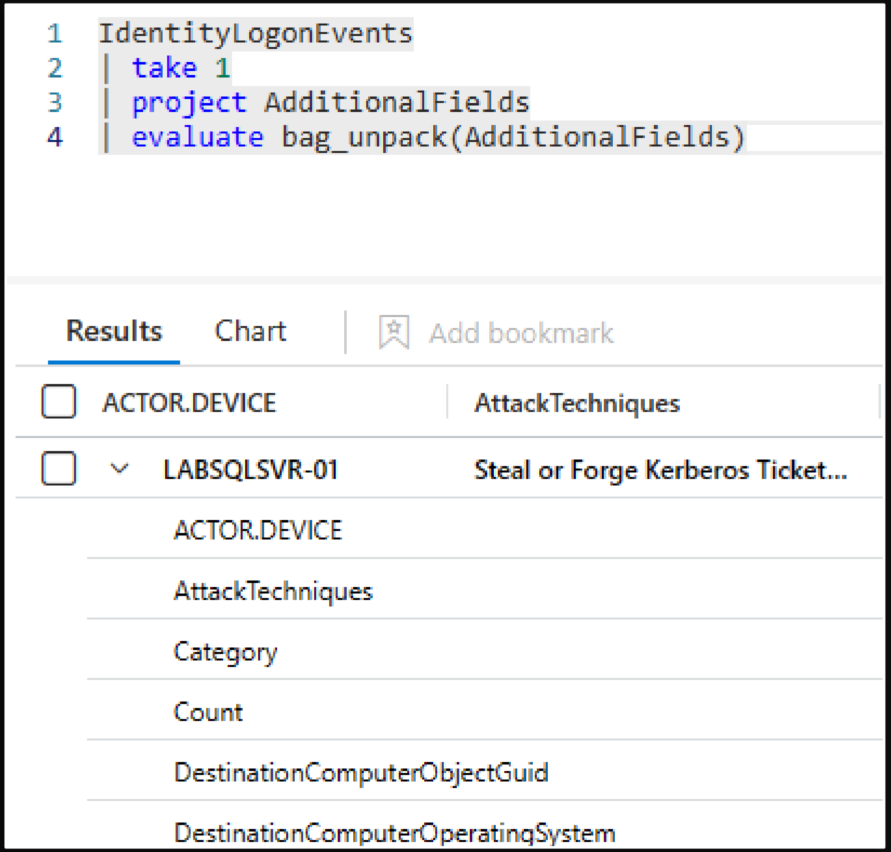
Syntax:

Table | evaluate bag_unpack (datatable)

Example:

IdentityLogonEvents

| evaluate bag_unpack(AdditionalFields)



```
IdentityLogonEvents
| take 1
| project AdditionalFields
| evaluate bag_unpack(AdditionalFields)
```

□ ACTOR.DEVICE	AttackTechniques
□ ⌄ LABSQLSVR-01	Steal or Forge Kerberos Ticket...

ACTOR.DEVICE

AttackTechniques

Category

Count

DestinationComputerObjectGuid

DestinationComputerOperatingSystem

Reference: https://learn.microsofthttps://learn.microsoft.com/en-us/kusto/query/bag-unpack-plugin

’bag_unpack’ example

```
DeviceFileEvents

| evaluate bag_unpack(AdditionalFields)
```

```
SigninLogs
| mv-expand todynamic(AuthenticationDetails)

| evaluate bag_unpack(AuthenticationDetails)
```

‘pivot’ plugin

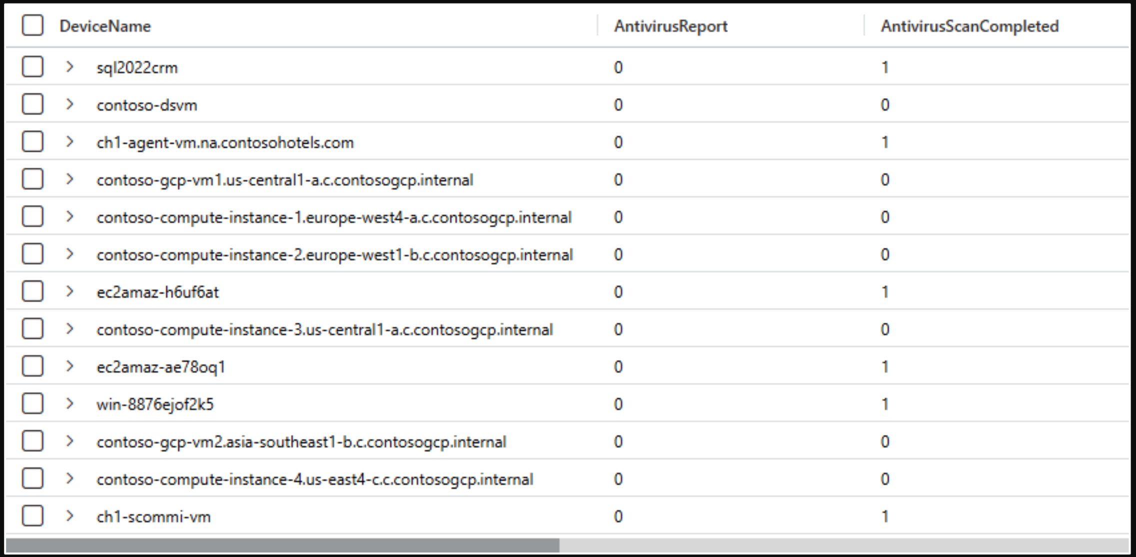
Rotates a table by turning the unique values from one column in the input table into multiple columns in the output table and performs aggregations as required on any remaining column values that will appear in the final output.

Syntax:

Table | evaluate pivot (pivotColumn, (aggregationFunction))

Examples:

```
DeviceEvents
| summarize count() by DeviceName, ActionType
| evaluate pivot (ActionType, sum(count_))
```



□	DeviceName	AntivirusReport	AntivirusScanCompleted
□	› sql2022crm	0	1
□	› contoso-dsvm	0	0
□	› ch1-agent-vm.na.contosohotels.com	0	1
□	› contoso-gcp-vm1.us-central1-a.c.contosogcp.internal	0	0
□	› contoso-compute-instance-1.europe-west4-a.c.contosogcp.internal	0	0
□	› contoso-compute-instance-2.europe-west1-b.c.contosogcp.internal	0	0
□	› ec2amaz-h6uf6at	0	1
□	› contoso-compute-instance-3.us-central1-a.c.contosogcp.internal	0	0
□	› ec2amaz-ae78oq1	0	1
□	› win-8876ejof2k5	0	1
□	› contoso-gcp-vm2.asia-southeast1-b.c.contosogcp.internal	0	0
□	› contoso-compute-instance-4.us-east4-c.c.contosogcp.internal	0	0
□	› ch1-scommi-vm	0	1

’pivot’ example

```
SigninLogs
| summarize count() by UserPrincipalName, ConditionalAccessStatus
| evaluate pivot(ConditionalAccessStatus, sum(count_))
```

Advanced String Manipulation

‘replace_string’ operator

Replaces all string matches with a specified string.

Syntax:

Table | replace_string(text, lookup, rewrite)

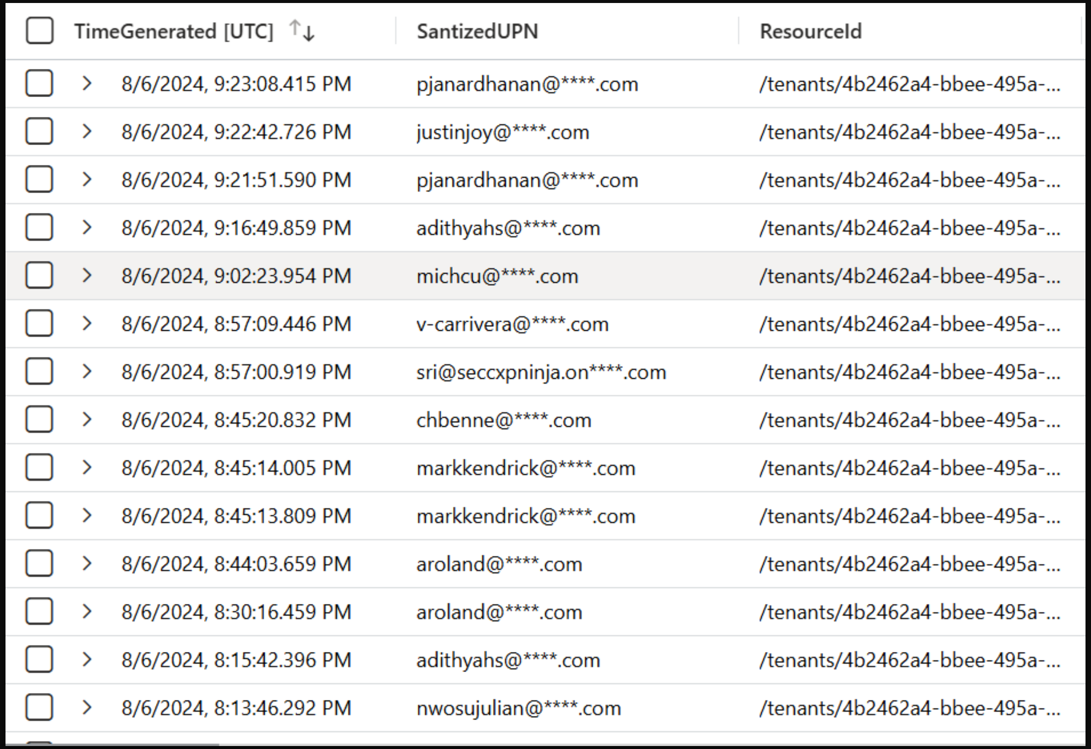
Example:

SigninLogs

| extend UserPrincipalName = replace_string(UserPrincipalName, "@contoso.com", "")

Tip:

Great for sanitizing PII from tables!



□		TimeGenerated [UTC] ↑↓	SantizedUPN	ResourceId
□	›	8/6/2024, 9:23:08.415 PM	pjanardhanan@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 9:22:42.726 PM	justinjoy@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 9:21:51.590 PM	pjanardhanan@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 9:16:49.859 PM	adithyahs@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 9:02:23.954 PM	michcu@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:57:09.446 PM	v-carrivera@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:57:00.919 PM	sri@seccxpninja.on****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:45:20.832 PM	chbenne@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:45:14.005 PM	markkendrick@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:45:13.809 PM	markkendrick@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:44:03.659 PM	aroland@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:30:16.459 PM	aroland@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:15:42.396 PM	adithyahs@****.com	/tenants/4b2462a4-bbee-495a-...
□	›	8/6/2024, 8:13:46.292 PM	nwosujulian@****.com	/tenants/4b2462a4-bbee-495a-...

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/replace-string-function

’replace_string’ example

```
SigninLogs
 | extend SantizedUPN = replace_string(UserPrincipalName,'microsoft','****’)
 | project-away UserPrincipalName
```

Regular Expressions (Regex) for KQL

‘extract’ operator

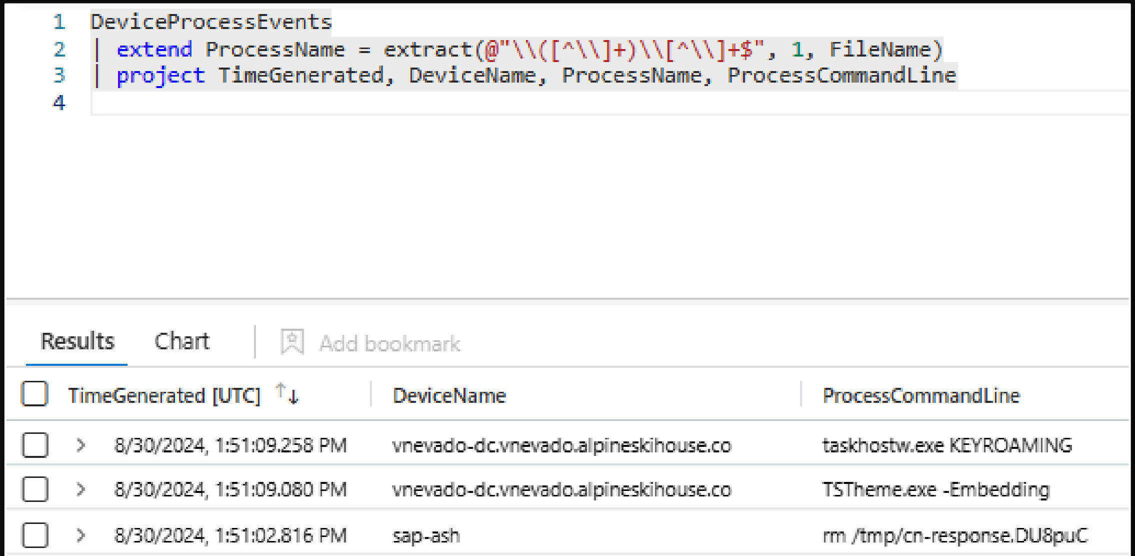
The ‘extract’ operator gets a match for a regular expression from a source string. Optionally, convert the extracted substring to the indicated type.

Syntax:

Table | extract(regex, captureGroup, source [, typeLiteral]))

Example:

DeviceProcessEvents
| extend ProcessName = extract(@"\\([^\\]+)\\[^\\]+$", 1, FileName)
| project TimeGenerated, DeviceName, ProcessName, ProcessCommandLine



```
DeviceProcessEvents
| extend ProcessName = extract(@"\\([^\\]+)\\[^\\]+$", 1, FileName)
| project TimeGenerated, DeviceName, ProcessName, ProcessCommandLine
```

□	TimeGenerated [UTC] ↑↓	DeviceName	ProcessCommandLine
□ >	8/30/2024, 1:51:09.258 PM	vnevado-dc.vnevado.alpineskihouse.co	taskhostw.exe KEYROAMING
□ >	8/30/2024, 1:51:09.080 PM	vnevado-dc.vnevado.alpineskihouse.co	TSTheme.exe -Embedding
□ >	8/30/2024, 1:51:02.816 PM	sap-ash	rm /tmp/cn-response.DU8puC

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/extract-function

’extract’ example

```
SecurityEvent
| where EventID == 4768
| take 10
| extend TicketOptions = extract(@"TicketOptions>(\S+?)<", 1, EventData)
| project TimeGenerated, Computer, TicketOptions
```

‘matches_regex’ operator

The ‘matches regex’ operator gets a record set based on a case-sensitive regex value.

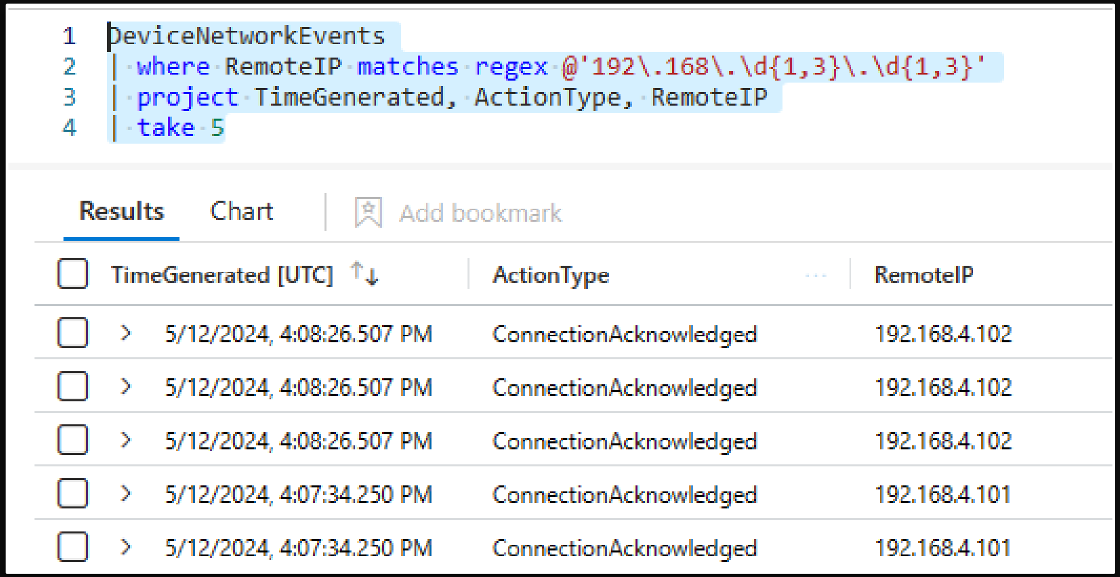
Syntax:

Table | where col matches regex (expression)

Example:

DeviceNetworkEvents

| where RemoteIP matches regex @ '192\.168\.\d{1,3}\.\d{1,3}'



```
DeviceNetworkEvents
| where RemoteIP matches regex @'192\.168\.\d{1,3}\.\d{1,3}'
| project TimeGenerated, ActionType, RemoteIP
| take 5
```

TimeGenerated [UTC]	ActionType	RemoteIP
5/12/2024, 4:08:26.507 PM	ConnectionAcknowledged	192.168.4.102
5/12/2024, 4:08:26.507 PM	ConnectionAcknowledged	192.168.4.102
5/12/2024, 4:08:26.507 PM	ConnectionAcknowledged	192.168.4.102
5/12/2024, 4:07:34.250 PM	ConnectionAcknowledged	192.168.4.101
5/12/2024, 4:07:34.250 PM	ConnectionAcknowledged	192.168.4.101

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/matches-regex-operator

’match_regex’ example

```
DeviceProcessEvents
| where TimeGenerated > ago(1h)
| where FileName matches regex @"^C:\\Windows\\.*\.exe$"
| project TimeGenerated, DeviceName, FileName, ProcessCommandLine
```

‘replace_regex’ operator

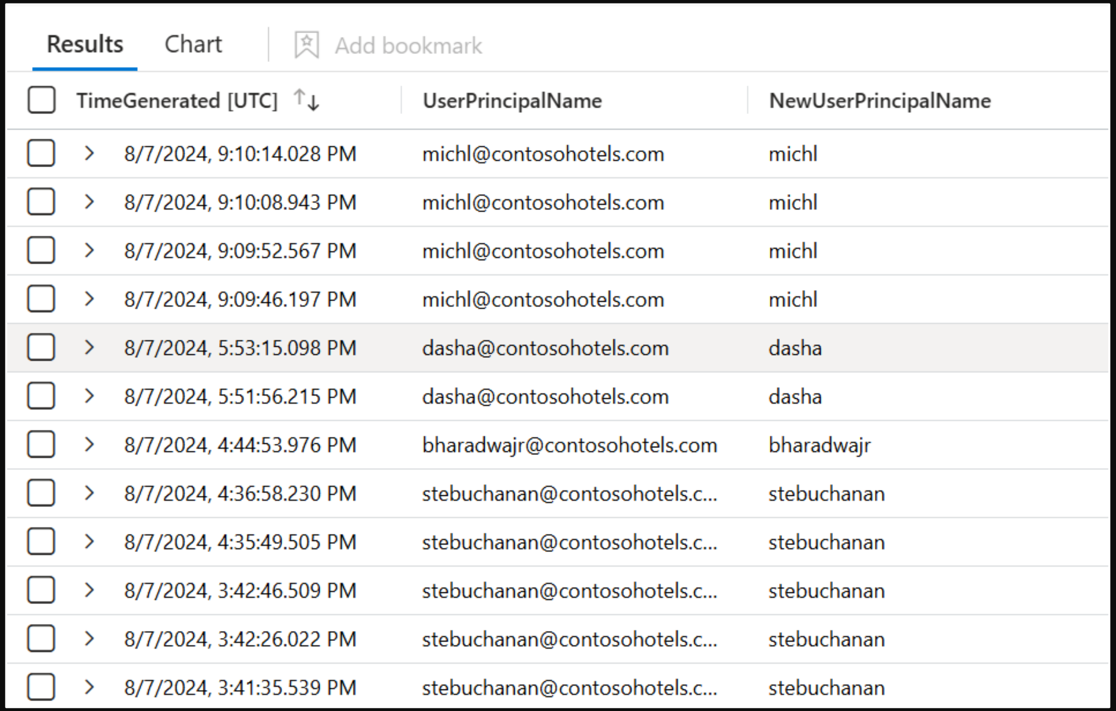
Replaces all regular expression matches with a specified pattern.

Syntax:

Table | replace_regex (source, lookup_regex, rewrite_pattern)

Example:

```
SigninLogs
| where UserPrincipalName contains “contosohotels.com“
| extend NewUserPrincipalName =
  replace_regex(UserPrincipalName, @"@contosohotels\.com$",
"")
```



□		TimeGenerated [UTC] ↑↓	UserPrincipalName	NewUserPrincipalName
□	›	8/7/2024, 9:10:14.028 PM	michl@contosohotels.com	michl
□	›	8/7/2024, 9:10:08.943 PM	michl@contosohotels.com	michl
□	›	8/7/2024, 9:09:52.567 PM	michl@contosohotels.com	michl
□	›	8/7/2024, 9:09:46.197 PM	michl@contosohotels.com	michl
□	›	8/7/2024, 5:53:15.098 PM	dasha@contosohotels.com	dasha
□	›	8/7/2024, 5:51:56.215 PM	dasha@contosohotels.com	dasha
□	›	8/7/2024, 4:44:53.976 PM	bharadwajr@contosohotels.com	bharadwajr
□	›	8/7/2024, 4:36:58.230 PM	stebuchanan@contosohotels.c...	stebuchanan
□	›	8/7/2024, 4:35:49.505 PM	stebuchanan@contosohotels.c...	stebuchanan
□	›	8/7/2024, 3:42:46.509 PM	stebuchanan@contosohotels.c...	stebuchanan
□	›	8/7/2024, 3:42:26.022 PM	stebuchanan@contosohotels.c...	stebuchanan
□	›	8/7/2024, 3:41:35.539 PM	stebuchanan@contosohotels.c...	stebuchanan

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/replace-regex-function

’replace_regex’ example

```
DeviceProcessEvents
| where InitiatingProcessFileName contains "netsh.exe"
| extend NewProcessName = replace_regex(InitiatingProcessFileName, @"\.exe$", "")
| project TimeGenerated, InitiatingProcessFileName, NewProcessName
```

Advanced Summarize Functions

‘case’ operator

Evaluates a list of conditions and returns the first result expression whose condition is satisfied.

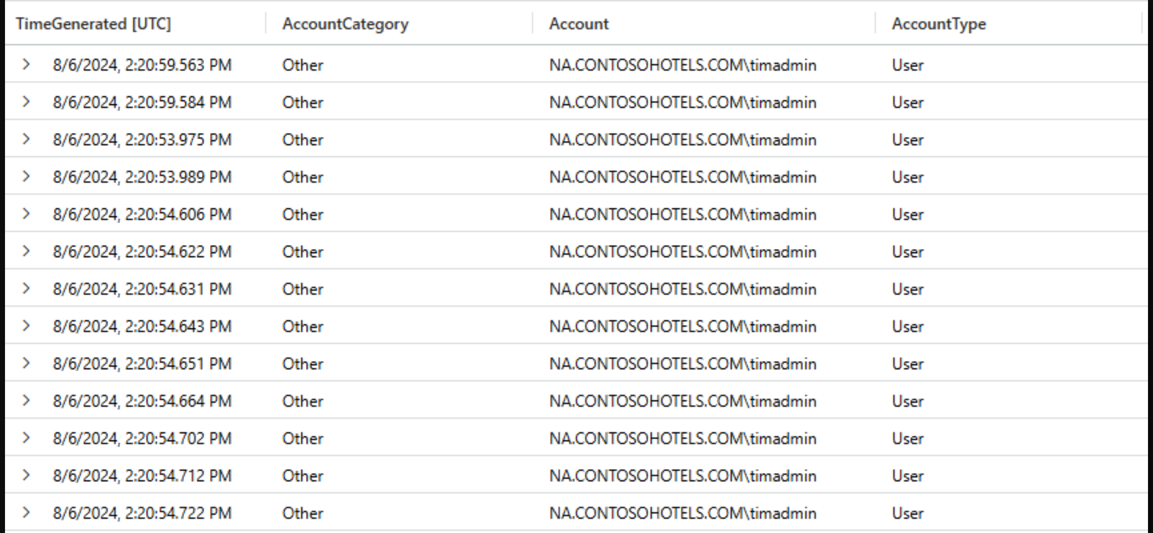
If none of the conditions return true, the result of the else expression is returned.

Syntax:

Table | case (<condition1>, <result1>, <condition2>, <result2>,..<default result>)

Example:

```
SecurityEvent
| where EventID == 4624
| where AccountType == ‘User’
| extend AccountCategory = case (TargetUserName startswith
"adm-","Administrative", TargetUserName startswith
"adm_”,"Service", "Other”)
```



	TimeGenerated [UTC]	AccountCategory	Account	AccountType
>	8/6/2024, 2:20:59.563 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:59.584 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:53.975 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:53.989 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.606 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.622 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.631 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.643 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.651 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.664 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.702 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.712 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User
>	8/6/2024, 2:20:54.722 PM	Other	NA.CONTOSOHOTELS.COM\timadmin	User

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/case-function

’case’ example

```
DeviceProcessEvents
| where TimeGenerated > ago(1d) // Filter events from the last 24 hours
| summarize count = count() by case(InitiatingProcessFileName == "explorer.exe", "Explorer Process",
                                    InitiatingProcessFileName == "svchost.exe", "Service Host Process",
                                    InitiatingProcessFileName == "chrome.exe", "Chrome Process",
                                    InitiatingProcessFileName == "winword.exe", "Word Process",
                                    "Other Process")
| order by count desc
```

‘make_list’ operator

The ‘make_list’ operator creates a dynamic array of all the values of expr in the group and returns a dynamic array of all the values of expr in the group.

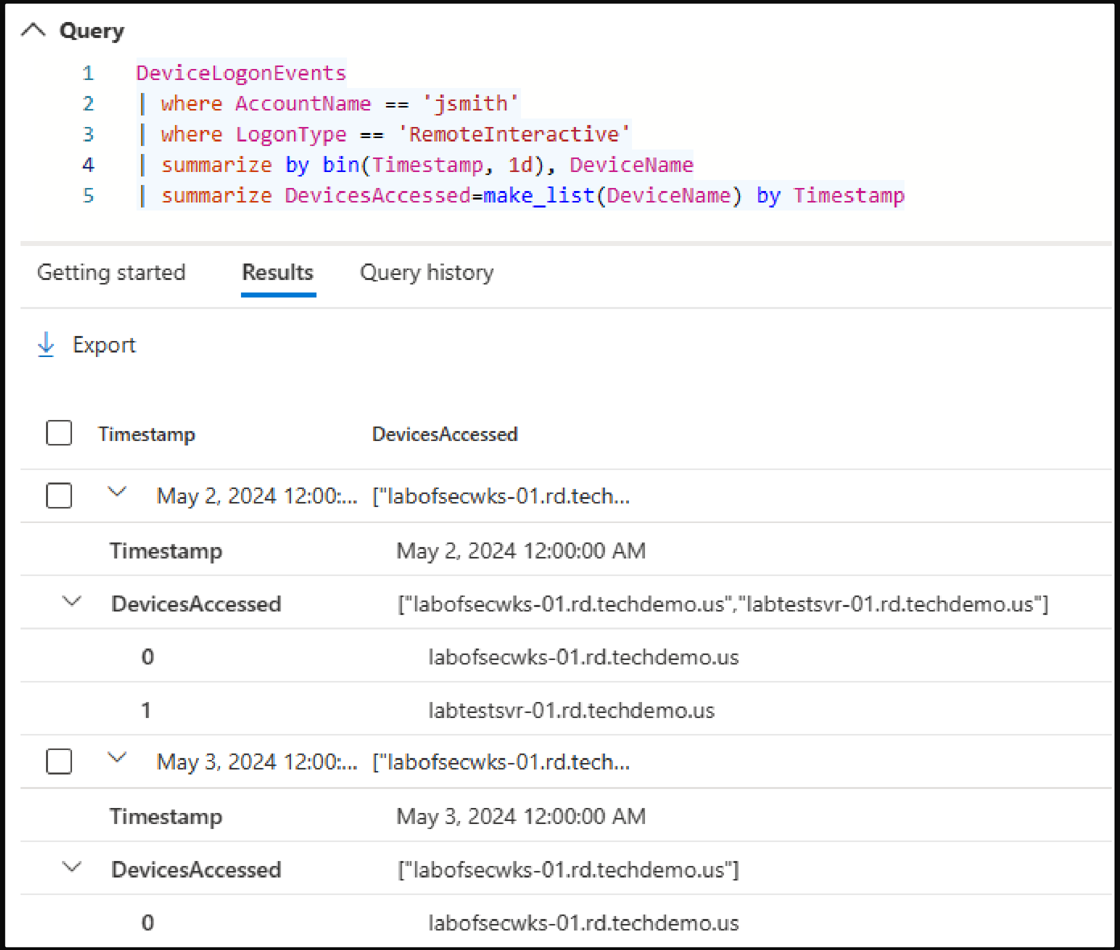
Syntax:

Table | summarize make_list(<Column>)

Example:

```
DeviceLogonEvents
| where LogonType == 'RemoteInteractive’
| summarize by bin(Timestamp, 1d), DeviceName
| summarize DevicesAccessed=make_list(DeviceName) by Timestamp
```

- If the input to the summarize operator isn't sorted, the order of elements in the resulting array is undefined.
- If the input to the summarize operator is sorted, the order of elements in the resulting array tracks that of the input.



```
DeviceLogonEvents
| where AccountName == 'jsmith'
| where LogonType == 'RemoteInteractive'
| summarize by bin(Timestamp, 1d), DeviceName
| summarize DevicesAccessed=make_list(DeviceName) by Timestamp
```

```
□   Timestamp                 DevicesAccessed
□ ˅ May 2, 2024 12:00:...    ["labofsecwks-01.rd.tech...
      Timestamp              May 2, 2024 12:00:00 AM
  ˅   DevicesAccessed        ["labofsecwks-01.rd.techdemo.us","labtestsvr-01.rd.techdemo.us"]
        0                      labofsecwks-01.rd.techdemo.us
        1                      labtestsvr-01.rd.techdemo.us
□ ˅ May 3, 2024 12:00:...    ["labofsecwks-01.rd.tech...
      Timestamp              May 3, 2024 12:00:00 AM
  ˅   DevicesAccessed        ["labofsecwks-01.rd.techdemo.us"]
        0                      labofsecwks-01.rd.techdemo.us
```

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/case-function

’make_list’ example

```
let sec_operators =
IdentityInfo
| where AssignedRoles contains "Security Operator"
| summarize make_list(AccountObjectId);
DeviceLogonEvents
| where AccountSid in (sec_operators)
```

‘make_set’ operator

The ‘make_set’ operator creates a dynamic array of all the values of expr in the group and returns a dynamic array of the set of distinct values that expr takes in the group:

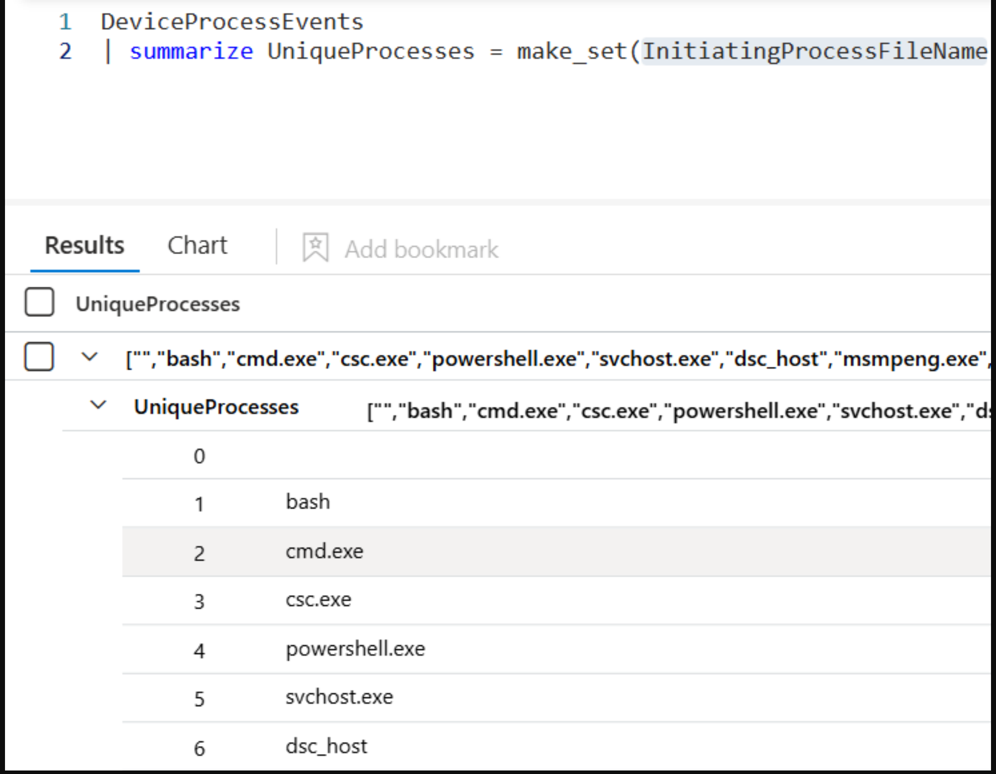
Syntax:

Table | summarize make_set(<Column>, #) by <Column>

Example:

DeviceProcessEvents

| summarize UniqueProcesses = make_set (InitiatingProcessFileName)



```
DeviceProcessEvents
| summarize UniqueProcesses = make_set(InitiatingProcessFileName
```

□	UniqueProcesses
□	["","bash","cmd.exe","csc.exe","powershell.exe","svchost.exe","dsc_host","msmpeng.exe",

```
UniqueProcesses     ["","bash","cmd.exe","csc.exe","powershell.exe","svchost.exe","d
        0
        1       bash
        2       cmd.exe
        3       csc.exe
        4       powershell.exe
        5       svchost.exe
        6       dsc_host
```

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/make-set-aggregation-function

’make_set’ example

```
let server2022_devices =
DeviceInfo
| where OSPlatform == 'WindowsServer2022'
| summarize make_set(DeviceName);
DeviceProcessEvents
| where DeviceName in (server2022_devices)
| where FileName =~ 'cmd.exe'
```

Advanced Time Filtering

‘set query_datetimescope_column’ operator

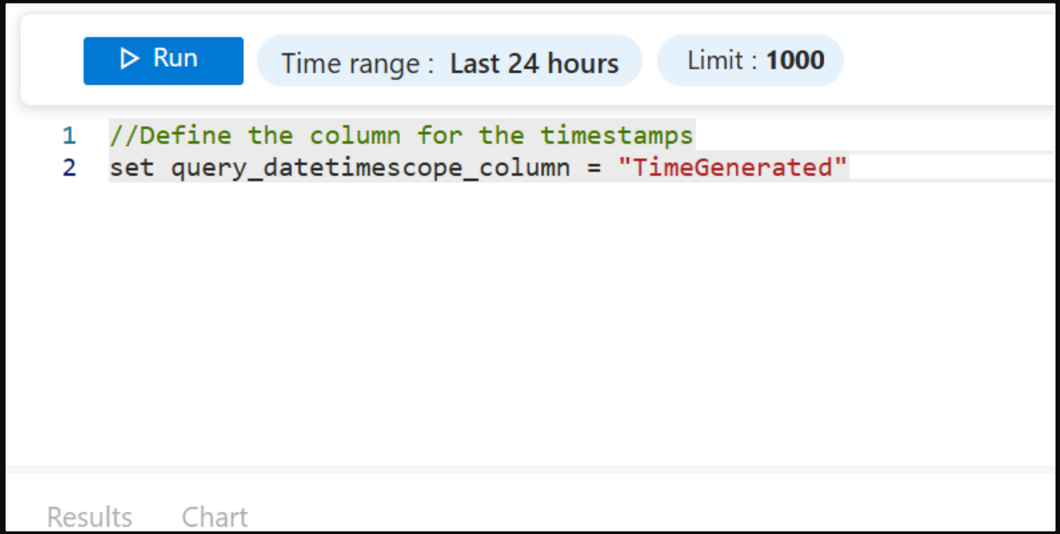
Specifies the column name for the query's datetime scope (query_datetimescope_to / query_datetimescope_from).

Syntax:

set *query_datetimescope_column* = <*TimeColumn*>

Example:

set query_datetimescope_column = “TimeGenerated”



```
//Define the column for the timestamps
set query_datetimescope_column = "TimeGenerated"
```

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/make-set-aggregation-function

‘set query_datetimescope_from’ operator

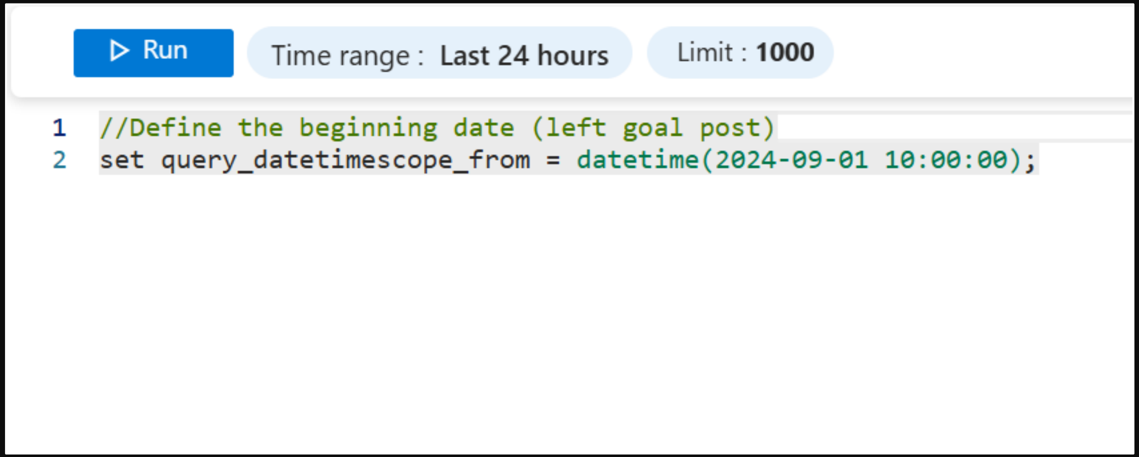
Sets the minimum date and time limit for the query scope. If defined, it serves as an auto-applied filter on query_datetimescope_column .

Syntax:

set *query_datetimescope_from = datetime(timestamp);*

Example:

set query_datetimescope_from = datetime(2024-09-01 10:10:00);

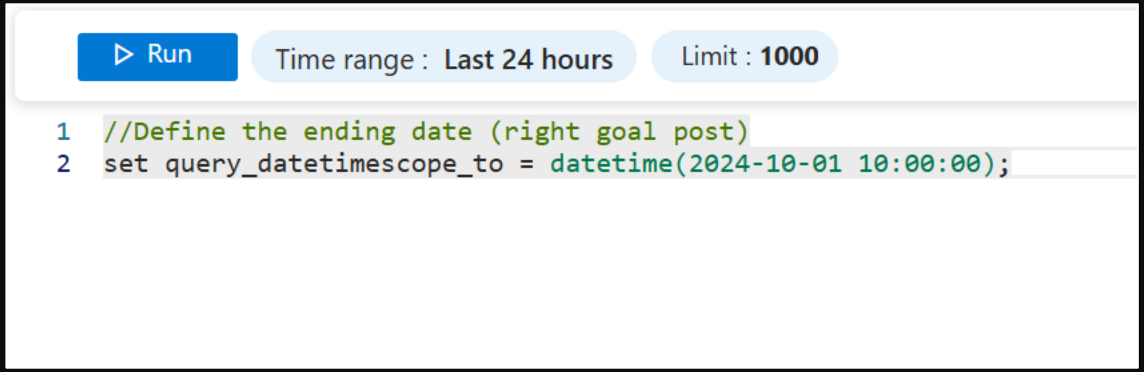


```
//Define the beginning date (left goal post)
set query_datetimescope_from = datetime(2024-09-01 10:00:00);
```

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/make-set-aggregation-function

‘set query_datetimescope_to’ operator

Sets the maximum date and time limit for the query scope. If defined, it serves as an auto-applied filter on query_datetimescope_column.



```
//Define the ending date (right goal post)
set query_datetimescope_to = datetime(2024-10-01 10:00:00);
```

Syntax:

set query_datetimescope_to = datetime(timestamp);

Example:

set query_datetimescope_to = datetime(2024-10-01 10:10:00);

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/api/rest/request-properties

Advanced Time Filtering example

```
set query_datetimescope_column = “TimeGenerated”;
set query_datetimescope_from = datetime(2024-07-01 10:10:00);
set query_datetimescope_to = datetime(2024-10-01 05:00:00);
```

Advanced Time Filtering

‘range’ operator

The ‘range’ operator A table with a single column called columnName, whose values are start, start + step, … up to and until stop:

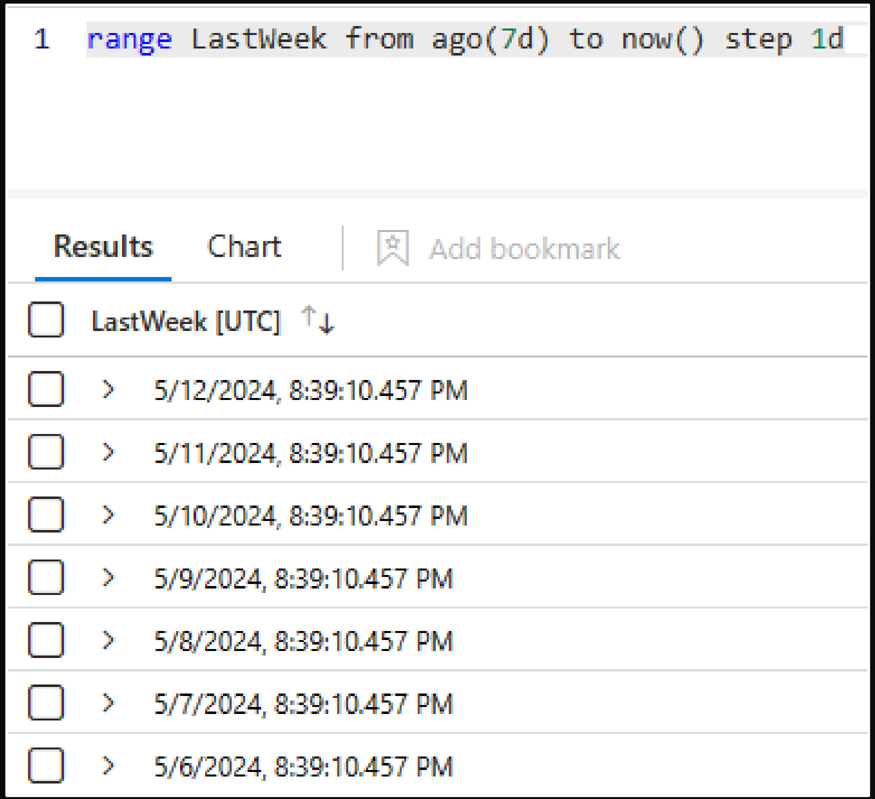
Syntax: range columnName from start to stop step bin

Example: range LastWeek from ago(7d) to now() step 1d

Start: The smallest value in the output.

Stop: The highest value being generated in the output or a bound on the highest value if step steps over this value.

Step: The difference between two consecutive values.



```
range LastWeek from ago(7d) to now() step 1d
```

LastWeek [UTC]
5/12/2024, 8:39:10.457 PM
5/11/2024, 8:39:10.457 PM
5/10/2024, 8:39:10.457 PM
5/9/2024, 8:39:10.457 PM
5/8/2024, 8:39:10.457 PM
5/7/2024, 8:39:10.457 PM
5/6/2024, 8:39:10.457 PM

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/range-operator

‘make-series’ operator

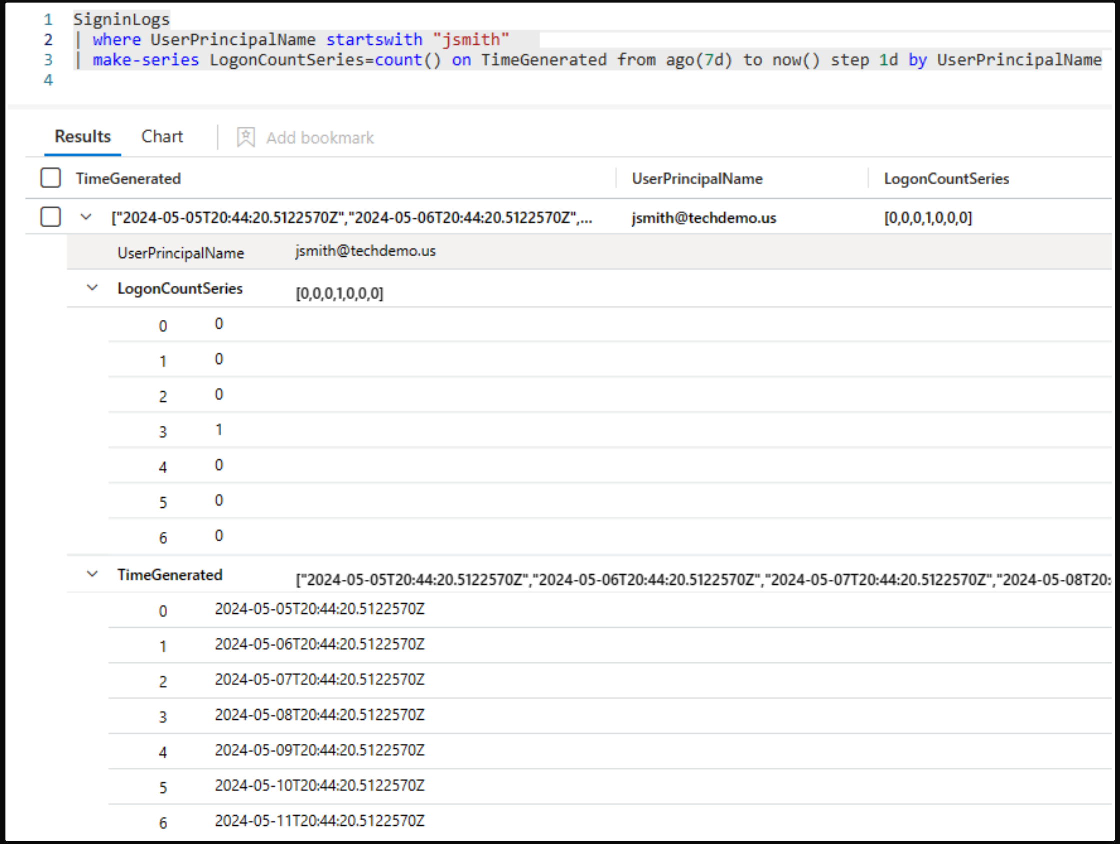
The ‘make-series’ operator creates a series of specified aggregated values along a specified axis:

Syntax:

Table
| make-series [MakeSeriesParameters] [Column =] Aggregation [default = DefaultValue] [, ...] on AxisColumn [from start] [to end] step step [by [Column =] GroupExpression [, ...]]

Example:

SigninLogs
| make-series LogonCountSeries=count() on TimeGenerated from ago(7d) to now() step 1d by UserPrincipalName



```
SigninLogs
| where UserPrincipalName startswith "jsmith"
| make-series LogonCountSeries=count() on TimeGenerated from ago(7d) to now() step 1d by UserPrincipalName
```

□ TimeGenerated	UserPrincipalName	LogonCountSeries
□ ["2024-05-05T20:44:20.5122570Z","2024-05-06T20:44:20.5122570Z",...	jsmith@techdemo.us	[0,0,0,1,0,0,0]

```
UserPrincipalName     jsmith@techdemo.us
LogonCountSeries      [0,0,0,1,0,0,0]
      0     0
      1     0
      2     0
      3     1
      4     0
      5     0
      6     0
TimeGenerated         ["2024-05-05T20:44:20.5122570Z","2024-05-06T20:44:20.5122570Z","2024-05-07T20:44:20.5122570Z","2024-05-08T20:
      0     2024-05-05T20:44:20.5122570Z
      1     2024-05-06T20:44:20.5122570Z
      2     2024-05-07T20:44:20.5122570Z
      3     2024-05-08T20:44:20.5122570Z
      4     2024-05-09T20:44:20.5122570Z
      5     2024-05-10T20:44:20.5122570Z
      6     2024-05-11T20:44:20.5122570Z
```

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/make-series-operator

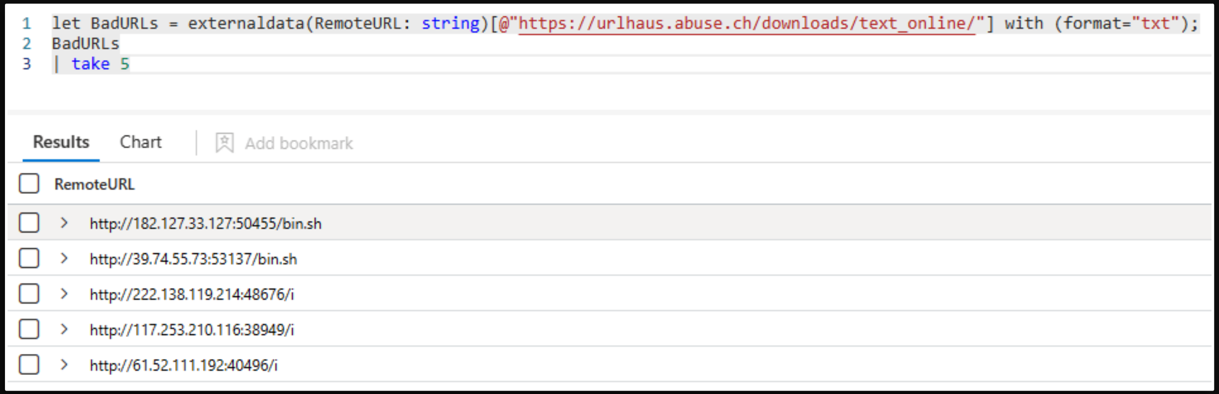
‘make-series’ example

```
SigninLogs
| make-series LogonCountSeries=count() on TimeGenerated from ago(7d) to now()
  step 1d by UserPrincipalName
```

External Data

‘externaldata’ operator

The ‘externaldata’ operator returns a table whose schema is defined in the query itself, and whose data is read from an external storage artifact, such as a blob in Azure Blob Storage or a file in Azure Data Lake Storage:



```
let BadURLs = externaldata(RemoteURL: string)[@"https://urlhaus.abuse.ch/downloads/text_online/"] with (format="txt");
BadURLs
| take 5
```

RemoteURL
http://182.127.33.127:50455/bin.sh
http://39.74.55.73:53137/bin.sh
http://222.138.119.214:48676/i
http://117.253.210.116:38949/i
http://61.52.111.192:40496/i

Syntax:

externaldata | (columnName:columnType [, ...]) [storageConnectionString [, ...]] [with (propertyName = propertyValue [, ...])]

Example:

```
SecurityEvent
| where Computer in ((externaldata (UserID:string) [
@"https://storageaccount.blob.core.windows.net/contoso/devices.txt" h@"?...SAS..." //Access Token provided by Azure]))
```

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/externaldata-operator

Query Across Log Analytics Workspaces

To reference another LAW workspace, you will have to use the workspace() expression.You can either use the Resource Name, GUID, Qualified Name, or the Azure Resource ID:

Resource Name (Easiest):

```
workspace("contosoretail“).Update | count
```

GUID:

```
workspace("b438b4f6-912a-46d5-9cb1-b44069212ab4“).Update | count
```

Qualified Name:

```
workspace("Contoso/ASC-Demo-RG/contosoretail“).Update | count
```

Azure Resource ID:

```
workspace( "/subscriptions/e427267-5645-4c4e-9c67-
3b84b59a6982/resourcegroups/ContosoAzureHQ/providers/Microsoft.OperationalInsights/workspaces/contosoretail“
).Event | count
```

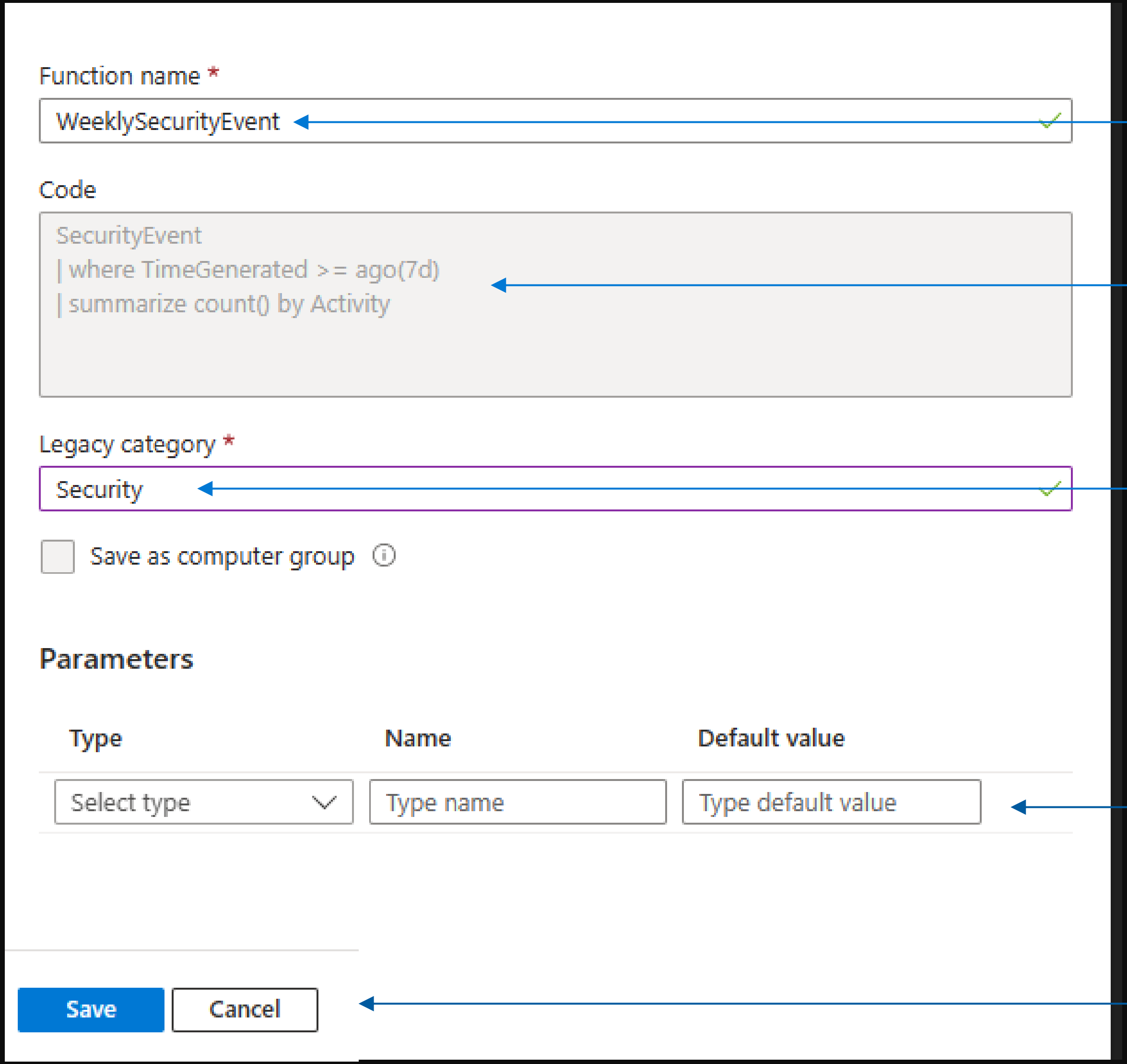
To query across multiple resource, you can use a union :

```
union Update, workspace("contosoretail-it“).Update, workspace("b459b4u5-912x-46d5-9cb1-p43069212nb4“).Update
```

Reference: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/externaldata-operator

Creating Shortcuts with Functions

The Anatomy of a Function:



Function name *

WeeklySecurityEvent

Code

```
SecurityEvent
| where TimeGenerated >= ago(7d)
| summarize count() by Activity
```

Legacy category *

Security

□ Save as computer group ⓘ

Parameters

Type	Name	Default value
Select type	Type name	Type default value

The Function Name

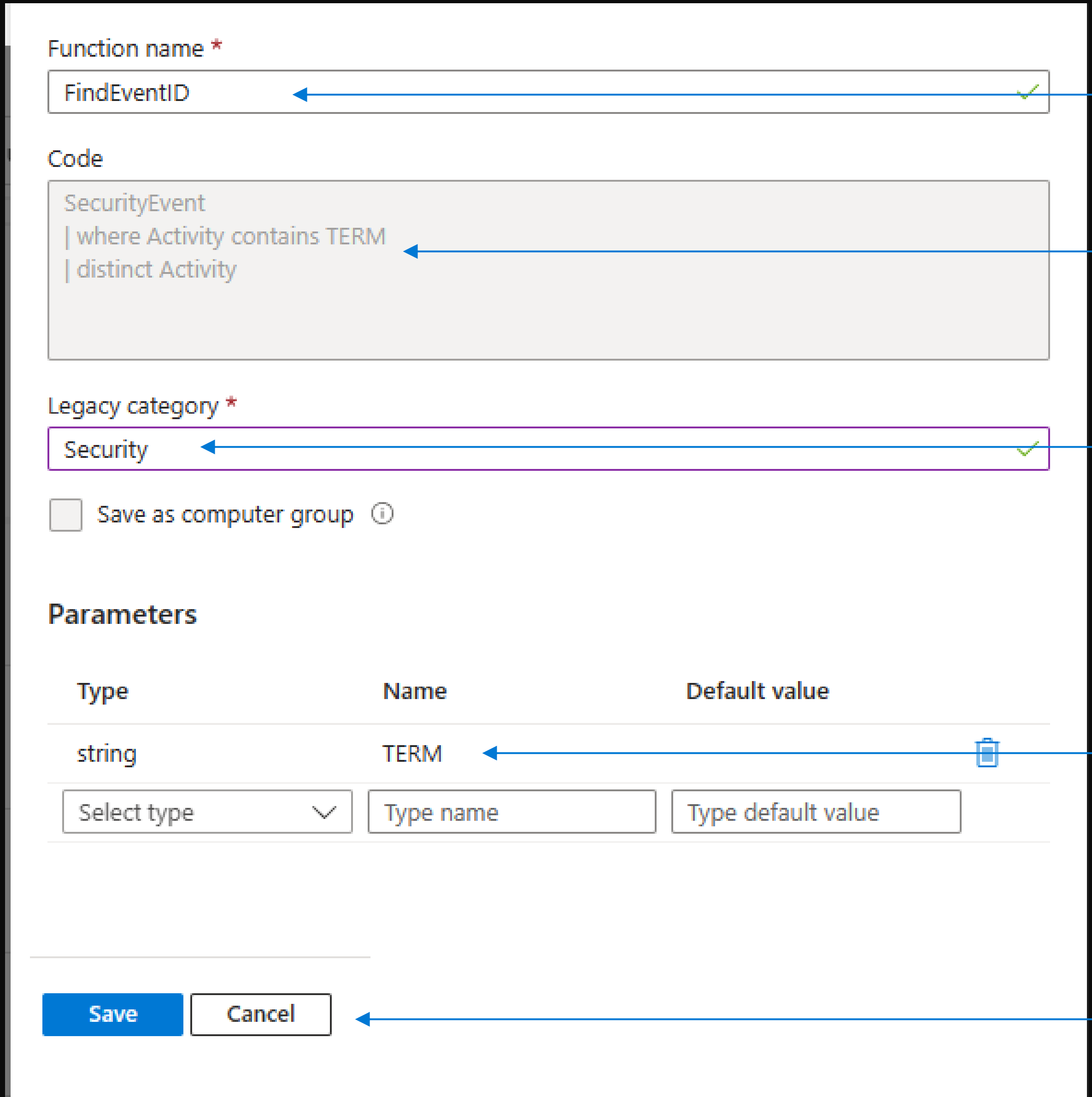
The Function Code

Unused category.
(Insert anything)

Function Parameters

Save / Cancel Buttons

The Anatomy of a Function with Parameters:



Function name *

FindEventID

Code

```
SecurityEvent
| where Activity contains TERM
| distinct Activity
```

Legacy category *

Security

□ Save as computer group

Parameters

Type	Name	Default value
string	TERM	
Select type	Type name	Type default value

The Function Name

The Function Code

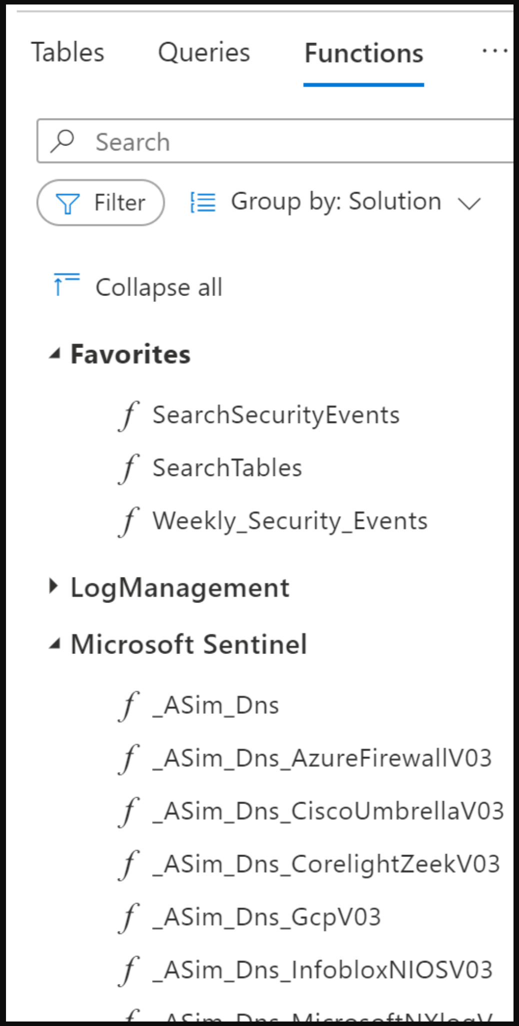
Unused category.
(Insert anything)

Function Parameter
(must match the
PARAM in code)

Save / Cancel Buttons

The Anatomy of a Function:

1. Give your function a purpose.
2. Create a query for the function in Logs.
3. Save the query as a function.
4. Add Parameters if needed.
5. Name and Save the function.



Tables Queries **Functions** ···

Search

Filter Group by: Solution

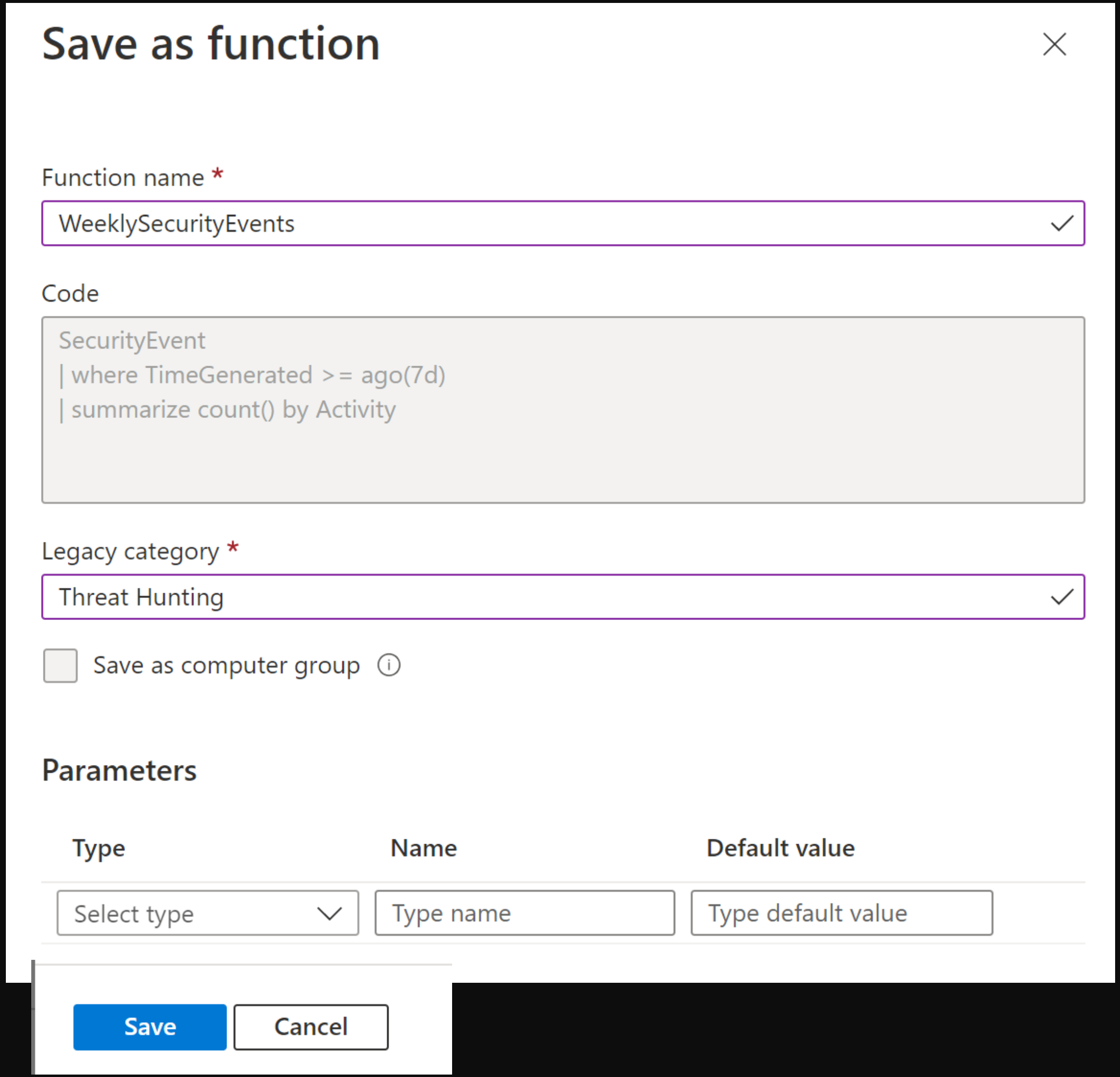
Collapse all

- **Favorites**
  - SearchSecurityEvents
  - SearchTables
  - Weekly_Security_Events
- **LogManagement**
- **Microsoft Sentinel**
  - _ASim_Dns
  - _ASim_Dns_AzureFirewallV03
  - _ASim_Dns_CiscoUmbrellaV03
  - _ASim_Dns_CorelightZeekV03
  - _ASim_Dns_GcpV03
  - _ASim_Dns_InfobloxNIOSV03
  - _ASim_Dns_MicrosoftNXlogV

Function 1: WeeklySecurityEvents

Query Code:

```
SecurityEvent
| where TimeGenerated >= ago(7d)
| summarize count() by Activity
```



Save as function

Function name *

WeeklySecurityEvents

Code

```
SecurityEvent
| where TimeGenerated >= ago(7d)
| summarize count() by Activity
```

Legacy category *

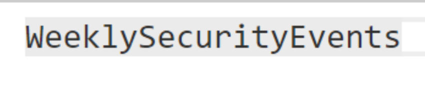
Threat Hunting

□ Save as computer group ⓘ

Parameters

Type	Name	Default value
Select type	Type name	Type default value

Example:



```
WeeklySecurityEvents
```

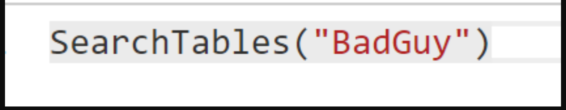
Function 2: SearchTables

Query Code:

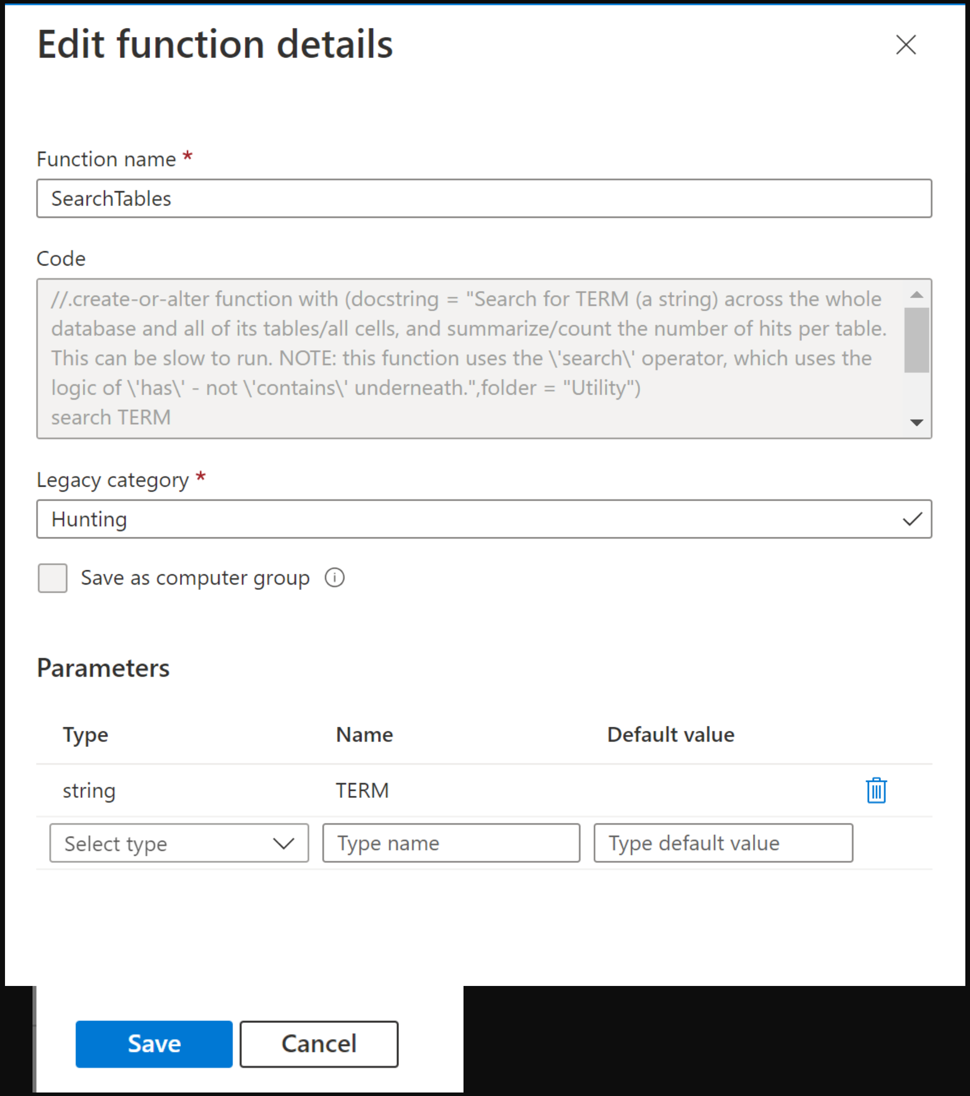
```
search TERM
    | summarize Count=count() by Table=$table
```

* Note the Parameter ‘TERM’ that is used.

Example:



```
SearchTables("BadGuy")
```



Edit function details

Function name *

SearchTables

Code

```
//.create-or-alter function with (docstring = "Search for TERM (a string) across the whole database and all of its tables/all cells, and summarize/count the number of hits per table. This can be slow to run. NOTE: this function uses the \'search\' operator, which uses the logic of \'has\' - not \'contains\' underneath.",folder = "Utility")
search TERM
```

Legacy category *

Hunting

□ Save as computer group

Parameters

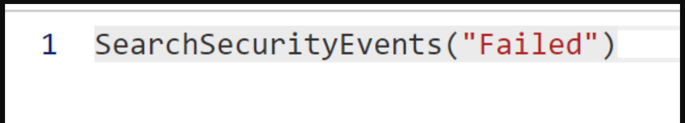
Type	Name	Default value
string	TERM	
Select type	Type name	Type default value

Function 3: SearchSecurityEvents

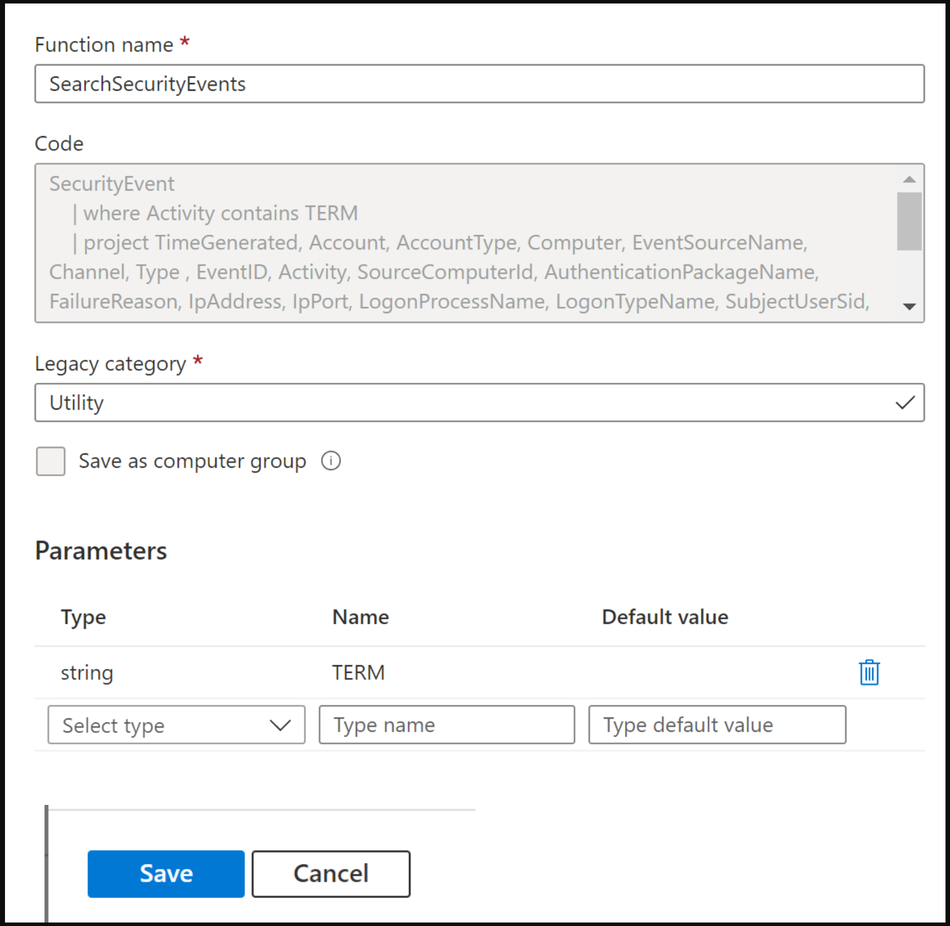
Query Code:

```
SecurityEvent
    | where Activity contains TERM
    | project TimeGenerated, Account ,Computer, Activity
```

Example:



```
SearchSecurityEvents("Failed")
```



Function name *

SearchSecurityEvents

Code

```
SecurityEvent
    | where Activity contains TERM
    | project TimeGenerated, Account, AccountType, Computer, EventSourceName,
Channel, Type , EventID, Activity, SourceComputerId, AuthenticationPackageName,
FailureReason, IpAddress, IpPort, LogonProcessName, LogonTypeName, SubjectUserSid,
```

Legacy category *

Utility

□ Save as computer group

Parameters

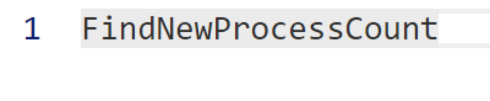
Type	Name	Default value
string	TERM	
Select type	Type name	Type default value

Function 4: FindNewProcessCount

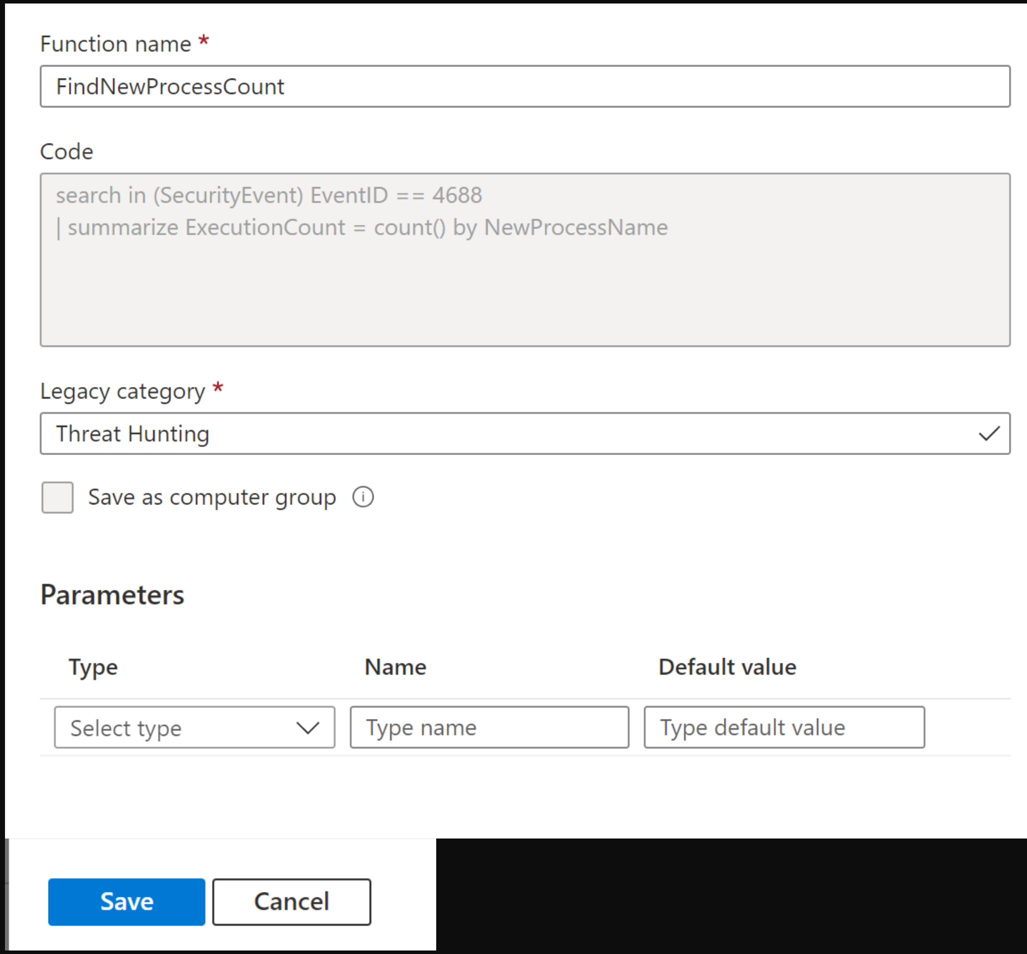
Query Code:

```
search in (SecurityEvent) EventID == 4688
| summarize ExecutionCount = count() by NewProcessName
```

Example:



```
FindNewProcessCount
```



Function name *

FindNewProcessCount

Code

```
search in (SecurityEvent) EventID == 4688
| summarize ExecutionCount = count() by NewProcessName
```

Legacy category *

Threat Hunting

□ Save as computer group ⓘ

Parameters

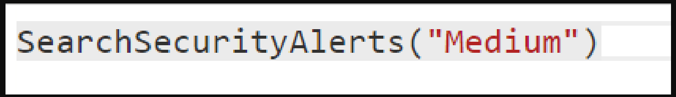
Type	Name	Default value
Select type	Type name	Type default value

Function 5: SearchSecurityAlerts

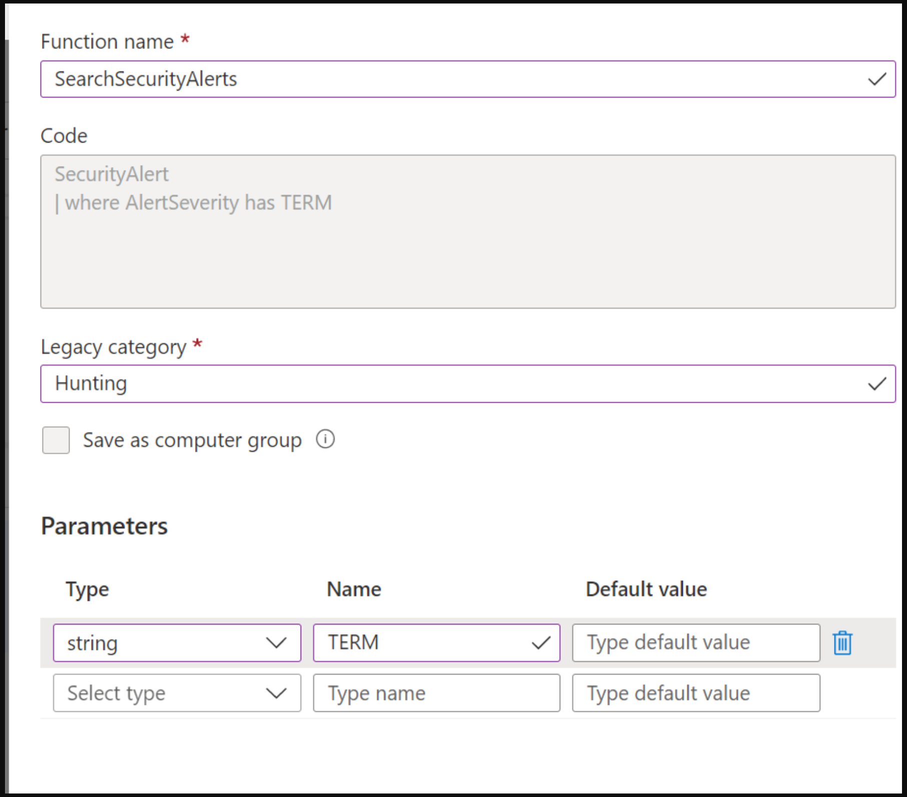
Query Code:

```
SecurityAlert
| where AlertSeverity has TERM
```

* Note the Parameter ‘TERM’ that is used.

Example: 

```
SearchSecurityAlerts("Medium")
```



Function name *

SearchSecurityAlerts ✓

Code

```
SecurityAlert
| where AlertSeverity has TERM
```

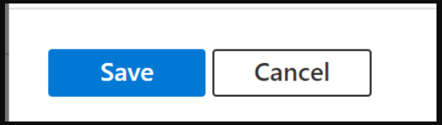
Legacy category *

Hunting ✓

□ Save as computer group ⓘ

Parameters

Type	Name	Default value
string	TERM ✓	Type default value
Select type	Type name	Type default value



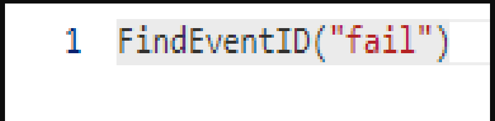
Function 6: FindEventID

Query Code:

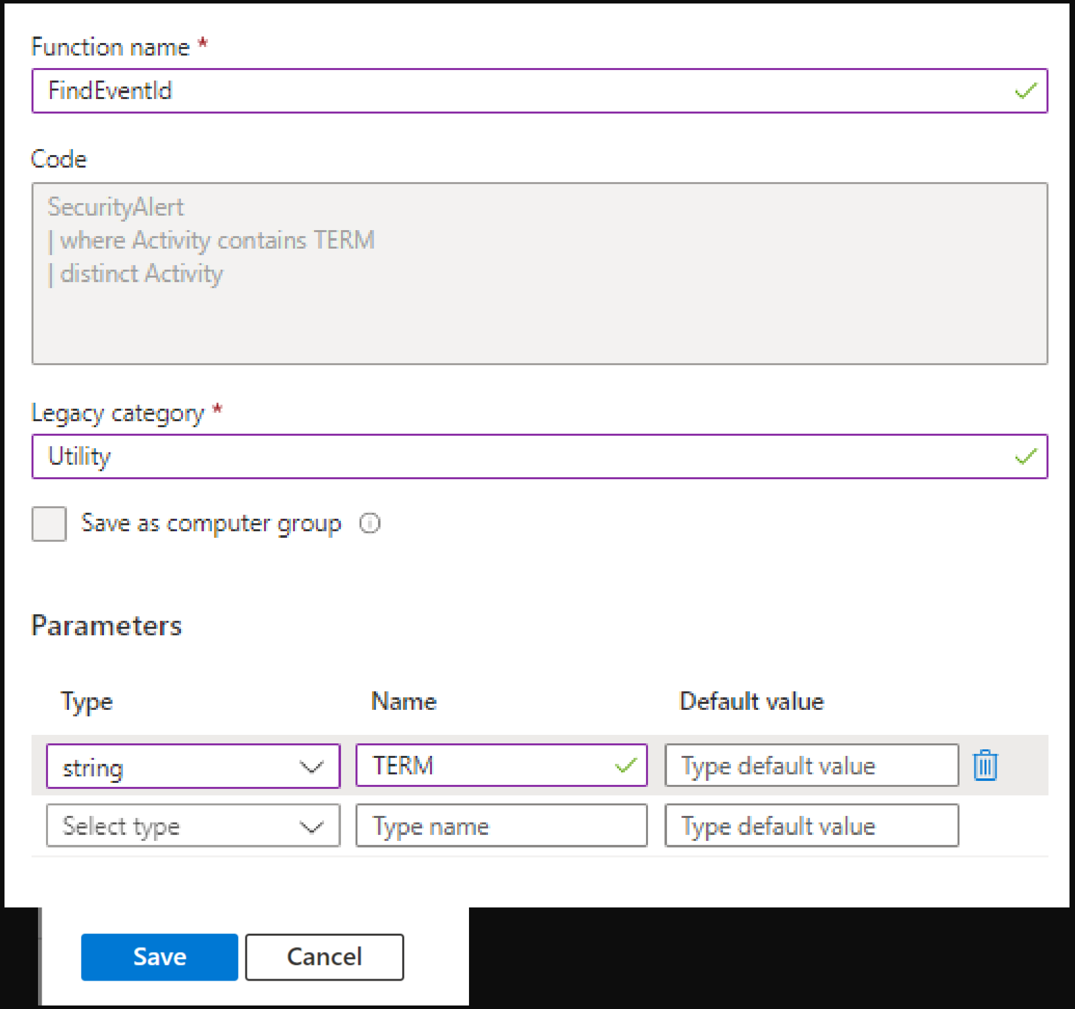
```
SecurityEvent
| where Activity contains TERM
| distinct Activity
```

* Note the Parameter ‘TERM’ that is used.

Example:



```
FindEventID("fail")
```



Function name * FindEventId

Code

```
SecurityAlert
| where Activity contains TERM
| distinct Activity
```

Legacy category * Utility

□ Save as computer group

Parameters

Type	Name	Default value
string	TERM	Type default value
Select type	Type name	Type default value