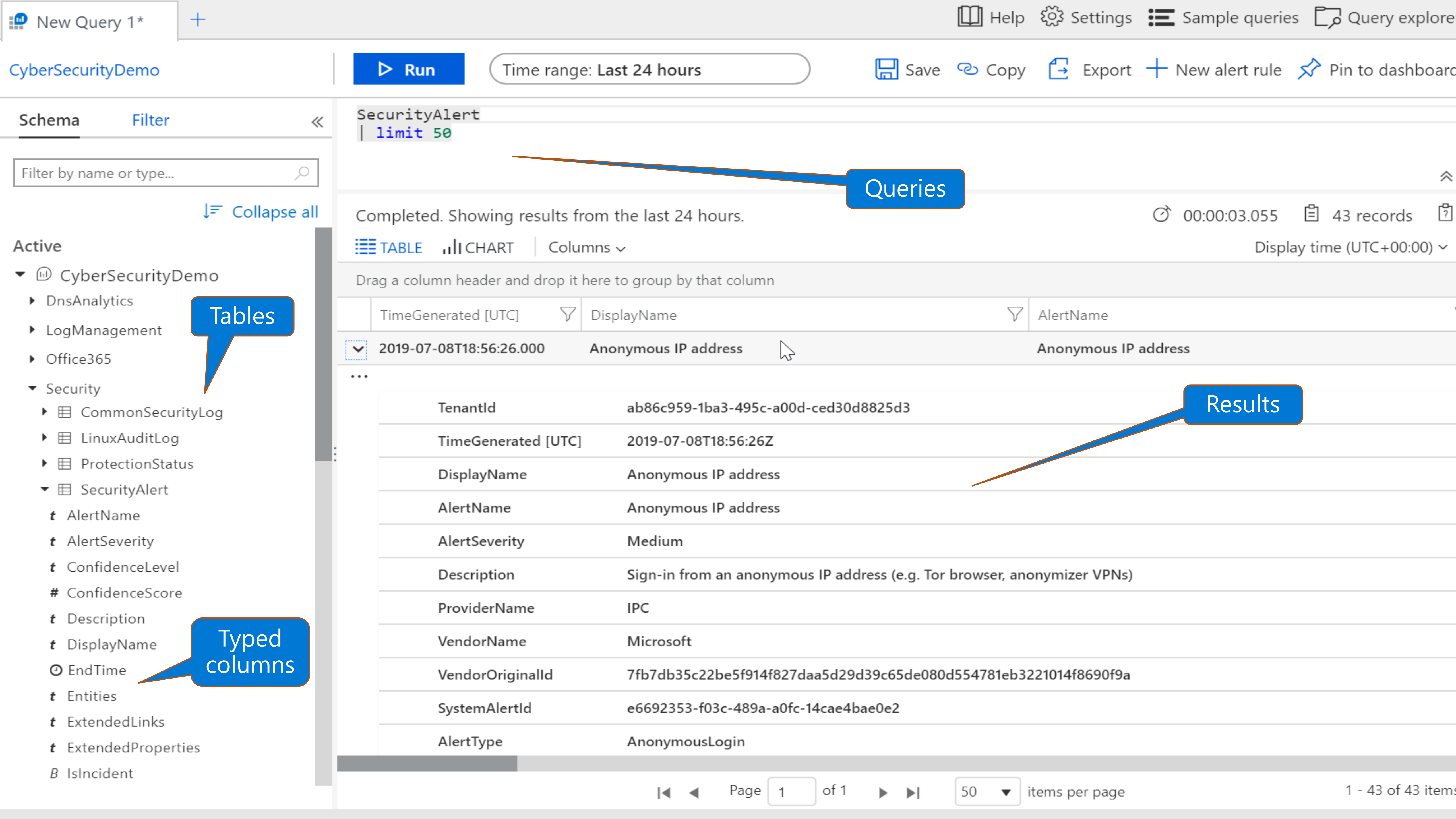
Azure Sentinel Level 400

KQL Workshop

Aka.ms/sentinelPerth



```
SecurityAlert
| limit 50
```

Completed. Showing results from the last 24 hours. 00:00:03.055 43 records

Active
- CyberSecurityDemo
  - DnsAnalytics
  - LogManagement
  - Office365
  - Security
    - CommonSecurityLog
    - LinuxAuditLog
    - ProtectionStatus
    - SecurityAlert
      - t AlertName
      - t AlertSeverity
      - t ConfidenceLevel
      - # ConfidenceScore
      - t Description
      - t DisplayName
      - EndTime
      - t Entities
      - t ExtendedLinks
      - t ExtendedProperties
      - B IsIncident

TimeGenerated [UTC]	DisplayName	AlertName
2019-07-08T18:56:26.000	Anonymous IP address	Anonymous IP address

TenantId	ab86c959-1ba3-495c-a00d-ced30d8825d3
TimeGenerated [UTC]	2019-07-08T18:56:26Z
DisplayName	Anonymous IP address
AlertName	Anonymous IP address
AlertSeverity	Medium
Description	Sign-in from an anonymous IP address (e.g. Tor browser, anonymizer VPNs)
ProviderName	IPC
VendorName	Microsoft
VendorOriginalId	7fb7db35c22be5f914f827daa5d29d39c65de080d554781eb3221014f8690f9a
SystemAlertId	e6692353-f03c-489a-a0fc-14cae4bae0e2
AlertType	AnonymousLogin

Page 1 of 1 · 50 items per page · 1 - 43 of 43 items

KQL Column Types

Basic

- **int, long**
- **bool**: true, false
- **string**: “example”, ‘example’

Time

- **datetime**: datetime(2016-11-20 22:30:15.4), now(), ago(4d)
- **timespan**: 2d, 20m, time(1.13:20:05.10), 100ms

Complex

- **dynamic**: JSON format

‘where’ command

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 “Kusto”*

- **String predicates**: ==, has, contains, startswith, endswith, matches regex, etc
- **Numeric/Date predicates**: ==, !=, <, >, <=, >=
- **Empty predicates**: isempty(), notempty(), isnull(), notnull()

‘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" // case insensitive
```

```
Perf
| where InstanceName matches regex "^[A-Z]:"
```

'limit' / 'take' command

Return up to the specified number of rows.

Syntax: *T | limit <number>*

Example: *SecurityEvent | limit 5*

- Sort is not guaranteed to be preserved.
- Consistent result is not guaranteed (when running the same query twice)
- Very useful when trying out new queries.
- Default limit is 10,000.

‘limit’ exercise

```
SecurityEvent
| limit 10
```

```
SecurityEvent
| where TimeGenerated > ago(1h)
| where EventID == 4624
| where AccountType =~ "user"
| take 10
```

‘count’ command

Returns the number of records in the input record set.

Syntax:	*T* | *count*
Example:	*SecurityEvent* | *count*

‘count’ exercise

```
Perf
| count

SecurityEvent
| where TimeGenerated > ago(1h)
| where EventID == 4624
| count
```

‘summarize’ command

Produces a table that aggregates the content of the input table.

Syntax:	*T | summarize Aggregation [by Group Expression]*
Examples:	*SecurityEvent | summarize count() by Computer*

- Simple aggregation functions: count(), sum(), avg(), min(), max(),
- Advanced functions (next slide): arg_min(), arg_max(), percentiles(), makelist(), countif()
- No Group Expression implies ‘distinct’.

‘summarize’ exercise

```
Perf
| where CounterName == "Free Megabytes"
| where InstanceName matches regex "^[A-Z]:$"
| summarize min(CounterValue)

SecurityEvent
| where TimeGenerated > ago(1h)
| where EventID == 4624
| summarize count() by AccountType, Computer
```

‘summarize’ advanced aggregations

arg_min(), arg_max(): returns the extreme value

Example: *Supplies | summarize arg_min(Price, Supplier) by Product*

// Cheapest supplier of each product

percentiles(): returns the value at the percentile

Example: *CallDetailRecords | summarize percentile(Duration, 95) by continent*

// The value of Duration that is larger than 95% of the sample set

makelist(), makeset(): returns a list of all values/distinct values respectively

Example: *PageViewLog | summarize countries=make_set(country) by continent*

‘summarize’ advanced aggregations exercise

```
SecurityEvent
| where EventID == 4624
| summarize arg_max(TimeGenerated, *) by Account

AzureDiagnostics
| summarize arg_max(TimeGenerated, *) by ResourceId

SecurityEvent | summarize
      AdminSuccessfullLogons =
            countif(Account contains "Admin" and EventID == 4624),
      AdminFailedLogons =
            countif(Account contains "Admin" and EventID == 4625)
```

Quiz #1

What is the difference between the following queries?

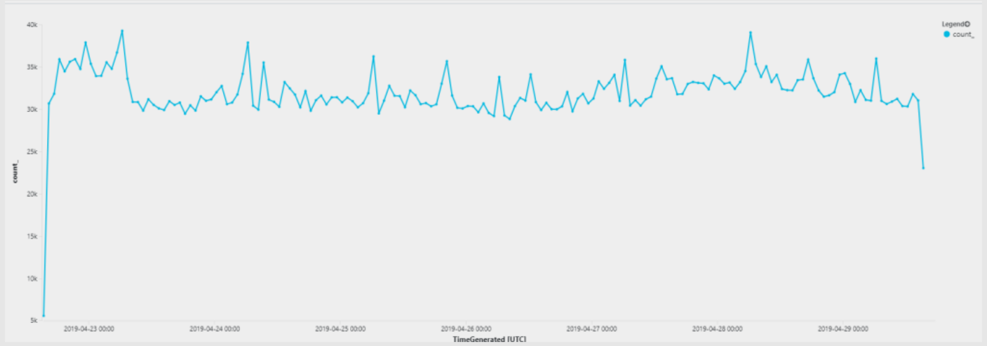
```
SecurityEvent
| summarize arg_max(TimeGenerated, *) by Account
| where EventID == "4624"
| count

SecurityEvent
| where EventID == "4624"
| summarize arg_max(TimeGenerated, *) by Account
| count
```

'summarize': bin and time series

A very useful summarize operation is creating time series:

SecurityEvent | summarize count() by bin(TimeGenerated, 1h) | render timechart



Other time measurements: 1h, 5d, 10m (defaults to 1h)

Can create multiple legends by aggregating additional field

‘bin’ exercise

```
SecurityEvent
| where TimeGenerated > ago(7d)
| summarize count() by bin(TimeGenerated, 1d)

Perf
| where CounterName == "Free Megabytes"
| where InstanceName matches regex "^[A-Z]:$"
| summarize min(CounterValue) by bin(TimeGenerated, 1d)
```

Question #1

Use the open to use KQL playground (https://aka.ms/LAdemo)

1. **Find how many times each process ran per computer** (hint: look for event ID that represent new process creation)

2. **Render a time chart of free MB over last 7 days for disk C:** (hint: InstanceName should be ‘C:’, CounterName should be ‘Free Megabytes’)

#1 solution

```
//Question 1
SecurityEvent
| where EventID == 4688
| summarize count() by NewProcessName, Computer

//Question 2
Perf
| where InstanceName == "C:"
| where CounterName == "Free Megabytes"
| summarize FreeMB = avg(CounterValue) by bin(TimeGenerated, 1h) | render
timechart
```

‘extend’ command

Create calculated columns and append them to the result set.

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

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

- The new added column is not indexed.
- To only change a column name, use ‘project-rename’.
- Useful function for in ‘extend’: iff, extract

‘project’ command

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

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

Example: *SecurityEvent | project TimeGenerated, Computer*

‘| project-away’ – Removed specified column/s.

‘| project-rename’ – Rename specified column/s.

‘extend’ & ‘project’ exercise

```
Perf
| where CounterName == "Free Megabytes"
| where InstanceName == "C:"
| extend FreeKB = CounterValue * 1000
| extend FreeGB = CounterValue / 1000

Perf
| where CounterName == "Free Megabytes"
| project Computer , CounterName , CounterValue

Perf
| where CounterName == "Free Megabytes"
| extend FreeKB = CounterValue * 1000
| extend FreeGB = CounterValue / 1000
| extend FreeMB = CounterValue
| project Computer , CounterName , FreeGB , FreeMB , FreeKB
```

‘distinct’ command

Produces a table with the distinct combination of the provided columns of the input table.

Syntax:	*T | distinct Column1, Column2*
Example:	*SecurityEvent | distinct Computer*

‘order by’ / ‘sort by’ & ‘top’ operator

Order by: Sort the rows of the input table into order by one or more columns.

Top: returns the top values after sort. Faster and can sort by expression

Syntax:	*T | sort by column [asc | desc] [nulls first | nulls last]*
	T | top NumberOfRows by Expression [asc | desc] [nulls first | nulls last]
Example:	*Table | order by country asc, price desc*

Don’t assume order by default

'order by' / 'top' exercise

```
SecurityEvent
| where TimeGenerated > ago(7d)
| order by TimeGenerated desc
| limit 100 // try also top

SecurityEvent
| top 100 by TimeGenerated desc

SecurityEvent
| where EventID == 4624
| summarize cnt=count() by Account
| top 10 by cnt
```

Question #2

Use the open to use KQL playground (https://aka.ms/LAdemo)

The table “SecurityEvent” contains security events collected from Windows machines. The events are identified by their IDs. Find the column that represents this ID, and complete the following tasks:

a. Render graph of logon events over time, starting from two weeks ago until one week ago.
b. Render graph of successful (4624) vs failed (4625) logons over the last 7 days, use alias for the legend (“Success”, “Failed”)
c. Find the last logon time for a user named “ContosoASCAlert\\LBecker”

#2

//Question #A: Render graph of logon events over time, starting from two weeks ago until one week ago.

```
SecurityEvent
| where EventID == 4624
| summarize count() by bin(TimeGenerated, 1h)
| render timechart
```

//Question #B: Render graph of successful (4624) vs failed (4625) logons over the last 7 days, use alias for the legend (“Success”, “Failed”)

```
SecurityEvent
| summarize Success=countif(EventID == 4624), Failed=countif(EventID==4625) by
bin(TimeGenerated, 1h)
| render timechart
```

//Question #C: Find the last logon time for a user named “ContosoASCAlert\\LBecker”

```
SecurityEvent
| where EventID == 4624
| where Account == "ContosoASCAlert\\LBecker"
| summarize max(TimeGenerated)
```

‘extract’ function

Get a match for a regular expression from a text string.

Syntax: *extract(regex, captureGroup, text [, typeLiteral])*

Example: *extract("x=([0-9.]+)", 1, "hello x=45.6|wo") == "45.6“*

‘let’ statement

Let statements bind names to expressions.

- declare global ‘variable’ or reuse of ‘variables’.
- Reuse of query runs
- Declare functions
- Declate dynamic table

`Example: See on demo.`

‘union’ operator

Takes two or more tables and returns the rows of all of them.

Example:
SecurityEvent | union (SecurityAlert | where Severity > 3)

- kind=inner(common columns), outer (all columns- default)
- Supports wildcard to union multiple tables (union Security*)
- Can union between tables from different clusters (or workspaces)

‘union’ exercise

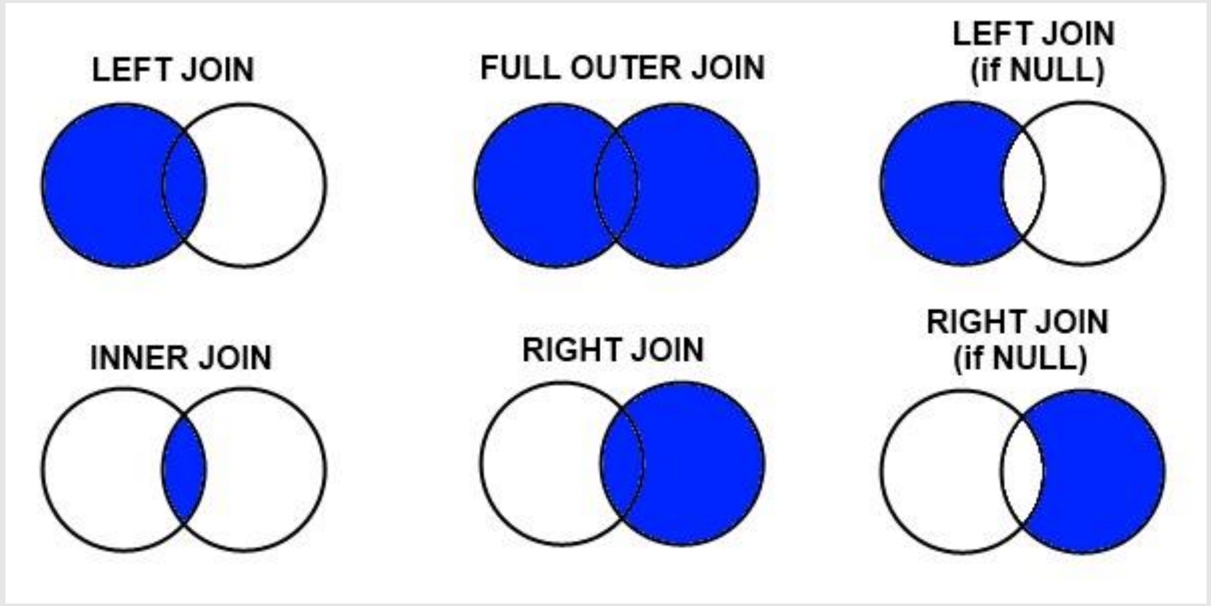
```
SecurityEvent
| union Heartbeat
| summarize count() by Computer
```

‘join’ operator

Merge the rows of two tables to form a new table by matching values of the specified column(s) from each table.

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

Example: *SecurityEvent | join (SecurityAlert | where Severity > 3) on Account*



Question #3

Find the ratio of alerts (in the SecurityAlert table) to events (in the SecurityEvent table)

#3 solution

```
SecurityAlert
| extend table = "SecurityAlers"
| union (SecurityEvent | extend table = "SecurityEvents")
| summarize SecurityAlerts = countif(table == "SecurityAlers") ,
SecurityEvents = countif(table == "SecurityEvents")
| extend Ratio = SecurityAlerts * 1.0 / SecurityEvents
| project SecurityEvents , SecurityAlerts , Ratio
```