SC-200 // KQL REFERENCE GUIDE

Schema-aware query construction, high-value operators, reusable hunts, joins, JSON, and speed rules.

RULE 01	RULE 02	RULE 03
Time first	Filter early	Know schema

01 // QUERY ANATOMY // TABLE | OPERATOR

```
let lookback = 24h;
DeviceProcessEvents
| where Timestamp > ago(lookback)
| where FileName in~ ("powershell.exe", "pwsh.exe")
| where ProcessCommandLine has_any ("DownloadString", "WebClient")
| summarize Runs=count(), Devices=dcount(DeviceId),
    First=min(Timestamp), Last=max(Timestamp) by AccountName
| where Runs >= 2
| order by Runs desc
```

Read left to right: select a table, reduce rows, shape columns, aggregate, rank, then render or operationalize. Each pipe receives the prior tabular result.

CORE OPERATORS // WHAT THEY DO

where	filter rows	`T | where predicate`
project	select/compute cols	`project A, B=X`
extend	append computed col	`extend X=expr`
summarize	aggregate/group	`summarize count() by X`
distinct	unique combinations	`distinct A, B`
top	sort + keep N	`top 10 by Count desc`
sort/order	order rows	`order by Time desc`
take/limit	sample/limit rows	`take 20`
join	match tables	`join kind=inner (...) on Key`
union	stack table rows	`union T1, T2`
mv-expand	array -> rows	`mv-expand Items`
parse	structure strings	`parse Col with ...`
render	visualize result	`render timechart`

MATCH SEMANTICS // EXAM FAVORITES

has	Indexed whole term. Prefer for words and better performance.
contains	Arbitrary substring. Broader and usually more expensive.
== / =~	Case-sensitive / case-insensitive equality.
in / in~	Case-sensitive / case-insensitive membership.
***_cs**	Case-sensitive string operator; narrower and often faster.

TIME + NULLS + TYPES

- Defender queries use UTC. Bound every hunt: Timestamp > ago(7d). Sentinel normally filters TimeGenerated.
- Bucket time with bin(TimeGenerated, 15m). Use between (start .. end) for a closed range.
- Cast explicitly: tostring(), toint(), todatetime(). Test with isempty(), isnotempty(), isnull(), and coalesce().
- let names a scalar, table, or function; terminate each let statement with a semicolon.

02 // SCHEMA FIRST // SENTINEL != DEFENDER

DEFENDER ADVANCED HUNTING // Timestamp	
DeviceProcessEvents	process creation and command lines
DeviceNetworkEvents	connections, remote IP/URL/port
DeviceFileEvents	file create/modify/rename/delete
DeviceRegistryEvents	registry activity
DeviceLogonEvents	device logon activity
IdentityLogonEvents	identity authentication activity
Email* / UrlClickEvents	mail, attachments, URLs, clicks
AlertInfo + AlertEvidence	alert metadata and implicated entities

SENTINEL / LOG ANALYTICS // TimeGenerated	
SecurityEvent	Windows Security log
SigninLogs / AuditLogs	Entra sign-ins and directory audit
Syslog	Linux/Unix events via AMA
CommonSecurityLog	CEF network/security events via AMA
AzureActivity	subscription control-plane operations
OfficeActivity	Microsoft 365 activity
SecurityAlert	alert context
SecurityIncident	case context

Syntax is shared; tables, columns, retention, and supported functions can differ. The live schema pane is authoritative.

QUERY RECIPE // HUNT WITH INTENT

1. State a falsifiable hypothesis and expected entity.
2. Choose the narrowest table and confirm column types.
3. Set UTC lookback first; apply selective filters early.
4. Project only evidence needed for the next pivot.
5. Summarize baseline/outlier; preserve raw examples.
6. Pivot by stable keys: DeviceId, AccountSid/UPN, SHA256, IP.
7. Validate benign causes; bookmark/save; map to ATT&CK.
8. Only then convert to analytics/custom detection logic.

SUMMARIZE // TURN EVENTS INTO SIGNAL

```
SecurityEvent
| where TimeGenerated > ago(24h) and EventID == 4625
| summarize Failures=count(), Hosts=dcount(Computer),
    First=min(TimeGenerated), Last=max(TimeGenerated) by Account
| where Failures > 20
| top 20 by Failures desc
```

High-value aggregators: count(), countif(), dcount(), min/max(), make_set(), make_list(), percentiles(), and arg_max(Time, *).

03 // JOINS + PIVOTS // KEEP THE ATTACK STORY

```
let Mail = EmailAttachmentInfo
| where Timestamp > ago(7d)
| project SHA256, RecipientEmailAddress, NetworkMessageId;
Mail
| join kind=inner (
    DeviceFileEvents
    | where Timestamp > ago(7d)
    | project SHA256, DeviceName, FileName, Timestamp
  ) on SHA256
| project Timestamp, RecipientEmailAddress, NetworkMessageId,
    DeviceName, FileName, SHA256
```

JOIN KIND // QUESTION WORDING -> SEMANTICS	
inner	All matching row pairs; preserves duplicates on the left.
innerunique	Default; deduplicates left-side keys before matching.
leftouter	All left rows plus any matches from the right.
leftanti	Only left rows with no right-side match; great for gaps.

Filter and project both sides before joining. In Defender guidance, keep the smaller input on the left. Rename keys before join when same-name columns are ambiguous.

DYNAMIC JSON + ARRAYS

```
DeviceEvents
| where Timestamp > ago(1d)
| where ActionType == "UsbDriveMount"
| extend af = parse_json(AdditionalFields)
| extend DriveLetter=tostring(af.DriveLetter)
| project Timestamp, DeviceName, DriveLetter

// one row per array element
T | mv-expand Item = todynamic(Items)
```

PERFORMANCE // MAKE THE QUERY CHEAP FIRST

- Name the table; avoid unscoped search or union. Search specific columns, not *.
- Apply time and selective predicates before parse, regex, summarize, join, or render.
- Prefer has to contains; prefer == / *_cs when case is known. Very short terms are not indexed.
- Project only needed columns before a join. Use count or take while developing.
- Prefer parse/parse_json to regex extraction when the format allows it.

DEBUG + EXAM CHECKLIST

- No rows? Check portal, table, time column/window, case, nulls, and ingestion latency.
- Too many? Tighten entity, action type, threshold, and exclusions; inspect raw rows before summarize.
- Wrong answer choices often use valid syntax with the wrong table/column or join semantics.
- Custom detection requires more than a hunt: compatible result schema, cadence, scope, and response actions.

SOURCES: [1] KQL quick ref [2] AH language [3] AH best practices [4] Study guide