

KQL for Security Analysts

SEC201

Learn the Kusto Query Language from your first query to mastery, against a month of security data with real attacks in it: filtering and shaping, aggregation, joins, parsing, functions, time series, graphs, and the checks that keep queries fast and answers right.

Modules	21
Lessons	128
Phases	5
Free preview	Course Orientation (8 lessons)

The method

Every query runs in the page. A practice lab built into each lesson holds the sample month in 21 tables with their real names and columns. Change a query, run it, and see what changed.

Predict, then run. Many queries ask for your prediction before they show their result, so every query is a small test of your understanding.

Three exercises in every lesson. Complete a query, fix a broken one, and write one from a blank editor, each graded on the result it returns. Each module ends with a knowledge check and a challenge set of questions the lessons did not answer.

Check every answer. The course treats an empty result, a surprising count and a borrowed or AI-written query as things to verify before anyone relies on them, and teaches the checks that do it.

Full syllabus

Orientation

Course Orientation [FREE PREVIEW](#)

KQL for Security Analysts: the query language of Microsoft Sentinel and Defender, taught against one month of an engineering company's records with several real attacks in them.

1. 0.1 What KQL Is and Where It Runs
2. 0.2 How Security Data Is Stored
3. 0.3 The Tables a Security Analyst Queries
4. 0.4 Sentinel Logs and Defender Advanced Hunting Compared
5. 0.5 Your First Query
6. 0.6 The Practice Lab and the Sample Month
7. 0.7 How to Learn KQL: the Mastery Ladder
8. 0.8 What This Course Builds

Foundation

Query Structure and Data Types

Module 1 of KQL for Security Analysts: how a KQL query is built from statements and a pipe of tables, how to explore a table and choose its columns, the ten scalar types, conversion, null and empty values, and a six-pass method for reading a query someone else wrote.

1. 1.1 Turning a Question into KQL
2. 1.2 Statements and Tabular Expressions
3. 1.3 The Pipe and Operator Order
4. 1.4 Learning a Table You Have Never Seen
5. 1.5 What a Value Really Is: Types and Conversion
6. 1.6 Incomplete Data: Nulls, Empties and Gaps in Coverage
7. 1.7 Checking a KQL Result Against Its Table
8. 1.8 Reading a Query You Did Not Write
9. Module Summary and Knowledge Check

Searching and Filtering

Module 2 of KQL for Security Analysts: finding where a value lives with search and find, narrowing a table with where, matching text the way the engine indexes it, working in time windows and around an event, and filtering addresses, nested fields and negations without losing rows.

1. 2.1 Searching for a Value When You Do Not Know the Column
2. 2.2 Narrowing a Table to the Events That Matter
3. 2.3 Matching Text the Way the Engine Indexes It
4. 2.4 Working Inside a Time Window
5. 2.5 Pivoting on a Timestamp: Windows Around an Event
6. 2.6 Filtering Addresses and Ranges
7. 2.7 Filtering Inside Nested Fields
8. 2.8 Negative Filters: !in, !has, not() and What They Silently Keep
9. Module Summary and Knowledge Check

Shaping Results

Module 3 of KQL for Security Analysts: deriving new values with extend, classifying rows with iff and case, cleaning text, working with time formats, choosing and naming columns, ordering and limiting, removing duplicates, and shaping a result for the person who will act on it.

1. 3.1 Deriving New Values: extend, Expressions and Rounding
2. 3.2 Classifying Rows: iff, case and Labels
3. 3.3 Cleaning and Normalizing Text
4. 3.4 Working with Time Values and Formats
5. 3.5 Choosing and Naming Columns: project and Its Variants
6. 3.6 Ordering and Limiting Results
7. 3.7 Removing Duplicates and Repeated Events
8. 3.8 Shaping a Result for Its Reader
9. Module Summary and Knowledge Check

Intermediate

Aggregation Functions

Module 4 of KQL for Security Analysts: how summarize builds a result, counting and distinct counting, describing distributions with medians and percentiles, conditional aggregates, lists and sets, binning activity over time, rates and denominators, and drawing aggregates as charts with render.

1. 4.1 From Rows to Groups: How summarize Builds a Result
2. 4.2 Counting Correctly: count, dcount and When an Estimate Matters
3. 4.3 Describing a Distribution: Typical, Unusual and Spread
4. 4.4 Several Questions in One Pass: Conditional Aggregates
5. 4.5 Collecting Evidence Inside a Group: Lists, Sets and Bags
6. 4.6 Activity Over Time: Bins, Rates per Period and the Busiest Window
7. 4.7 Rates Done Right: Denominators and Averages of Averages
8. 4.8 Showing an Aggregate: render and When a Chart Beats a Table
9. Module Summary and Knowledge Check

Multi-Table Queries

Module 5 of KQL for Security Analysts: stacking tables with union, matching rows with join, the default innerunique join, outer, anti and semi joins, lookup and reference tables, joining events across time, and checking that a join returned the right answer.

1. 5.1 Putting Sources Side by Side: union
2. 5.2 Matching Rows Across Tables: How join Works
3. 5.3 The Default Join Drops Rows: innerunique against inner
4. 5.4 Keeping Rows That Have No Match: Outer Joins
5. 5.5 What One Table Has That Another Lacks: Anti and Semi Joins
6. 5.6 Adding Context to Rows: lookup and Reference Tables
7. 5.7 Joining Across Time: Events Near Each Other
8. 5.8 Joins That Return the Wrong Answer
9. Module Summary and Knowledge Check

String and Structured Data

Module 6 of KQL for Security Analysts: pulling fields out of free text with parse, key-value logs, regular expressions and extract, indicators and encoded commands in text, JSON and the dynamic type, property bags, and turning arrays into rows with mv-expand and mv-apply.

1. 6.1 Pulling Fields out of Free Text: parse
2. 6.2 Text That Varies: parse-where and Key-Value Logs
3. 6.3 Matching Patterns: Regular Expressions and extract
4. 6.4 Indicators Buried in Text: extract_all and Decoding
5. 6.5 Nested Data: JSON and the dynamic Type
6. 6.6 Property Bags and Changing Schemas
7. 6.7 One Row per Element: mv-expand
8. 6.8 Working Inside an Array: mv-apply
9. Module Summary and Knowledge Check

Advanced

Variables, Functions and External Data

Module 7 of KQL for Security Analysts: naming values and intermediate results with let, materialize, writing functions, parsers as functions, watchlists, externaldata, matching indicators in every format, and IP address functions and ranges.

1. 7.1 Naming Values and Thresholds: let for Scalars
2. 7.2 Naming Intermediate Results: let for Tables
3. 7.3 Writing Your Own Functions
4. 7.4 Parsers as Functions: Saved and Shared
5. 7.5 Lists Kept Outside the Query: Watchlists
6. 7.6 Data from Outside the Workspace: externaldata
7. 7.7 Indicators in Every Format
8. 7.8 Address Functions and Ranges
9. Module Summary and Knowledge Check

Time-Series Analysis and Window Functions

Module 8 of KQL for Security Analysts: make-series and the zeros it keeps, series arithmetic and statistics, reading series function output, comparing periods, series per entity, row order with serialize, prev and next for sessions, and running totals.

1. 8.1 Counts Over Time That Include the Zeros: make-series
2. 8.2 Working on Whole Series: Arithmetic, Gaps and Statistics
3. 8.3 Reading What a Series Function Returns
4. 8.4 This Week Against Last: Comparing Periods
5. 8.5 One Series per Entity
6. 8.6 Order in a Table: serialize and Row Numbers
7. 8.7 The Row Before and After: prev, next and Sessions
8. 8.8 Running Totals and Cumulative Counts
9. Module Summary and Knowledge Check

Graph Operators

Module 9 of KQL for Security Analysts: relationships as nodes and edges, make-graph, graph-match patterns, variable-length paths with map, all, any and inner_nodes, process trees, identity paths, machine-to-machine connections, and when a join is the better tool.

1. 9.1 Relationships as a Graph
2. 9.2 Building a Graph: make-graph
3. 9.3 Finding Patterns: graph-match
4. 9.4 Paths of Any Length
5. 9.5 Process Trees as Graphs
6. 9.6 Who Can Reach What: Identity Paths
7. 9.7 Connections Between Machines
8. 9.8 Graph or Join: Choosing
9. Module Summary and Knowledge Check

Mastery

Query Optimization and Validation

Module 10 of KQL for Security Analysts: how a query runs, filtering early, the cost of string operators, joins and summaries at scale, documented limits, adapting public queries, checking answers for silent zeros and data gaps, and checking AI-written KQL.

1. 10.1 How a Query Runs
2. 10.2 Filter First: Time, Table and Columns
3. 10.3 String Operators and Their Cost
4. 10.4 Joins and Summaries at Scale
5. 10.5 Limits in Sentinel and Advanced Hunting
6. 10.6 Reading and Adapting Public Queries
7. 10.7 Checking the Answer
8. 10.8 Checking AI-Written KQL
9. Module Summary and Knowledge Check

Phase 0: Course Resources

Cheatsheets

One sheet per module of KQL for Security Analysts, each entry a question, a query that answers it against the sample month, what to read and what the answer does not prove.

1. Tables and the Lab
2. Query Structure and Types
3. Searching and Filtering
4. Shaping Results
5. Aggregation
6. Multi-Table Queries
7. String and Structured Data
8. Variables, Functions and External Data
9. Time Series and Window Functions
10. Graph Operators
11. Query Optimization and Validation

Cookbooks

Procedures for the three problems every KQL user meets: a query that returns nothing, a query that is too slow, and a log source nobody has parsed.

1. A Query Returns Nothing
2. A Query Is Too Slow
3. Parsing an Unfamiliar Log Source

Lab Setup

The course's practice lab, which needs no setup, and how to get a workspace of your own to run the course's queries and the Project against.

1. The Practice Lab
2. Your Own Workspace

Walkthroughs

Six investigations worked end to end with the output at every step, five of them turning on a query that ran cleanly and answered wrongly.

1. Building a Query From a Question
2. When the Join Lies
3. The Baseline That Learned the Attack
4. The Query That Timed Out
5. The Parse That Half Worked
6. Reconstructing the Process Tree

Query Patterns

Five query patterns that recur across security questions: first time seen, rare in population, volume anomaly per entity, paired events in sequence, and regularity.

1. First Time Seen
2. Rare in Population
3. Volume Anomaly per Entity
4. Paired Events in Sequence
5. Beaconsing and Regularity

Playground

Resources for practicing KQL and using it at work: the detection library, a free-run query console, guided investigations, response playbooks, and an open-source DFIR toolkit.

Operational Reference

Every operator, function, check and limit from KQL for Security Analysts in one place, organized by the task in front of you and linked back to the section that explains it.

1. Operational Reference

References & Further Reading

The documentation behind KQL for Security Analysts, arranged by the question it answers: the language, where queries run and their limits, getting data, and community query material.

Project

Project: Your Own Query Library

The KQL for Security Analysts project: a brief for a library of twenty queries you build on your own data, each answering a stated question, each checked, with its table's pitfalls and its cost noted, and one taken to the quality of a scheduled rule.

1. The Brief

Course Completion

Course Exam

KQL for Security Analysts end-of-course exam: twenty tasks, two per teaching module, each answered by writing KQL against the course's sample month and checked on the server.

1. Course Completion. KQL for Security Analysts

Generated 2026-10-02 from the published course. The current syllabus is always at ridgelinecyber.com/training/courses/kql-for-security-analysts/