u2u

# Azure Log Analytics and the Kusto Query Language

Collect, Query and Analyze Your Data

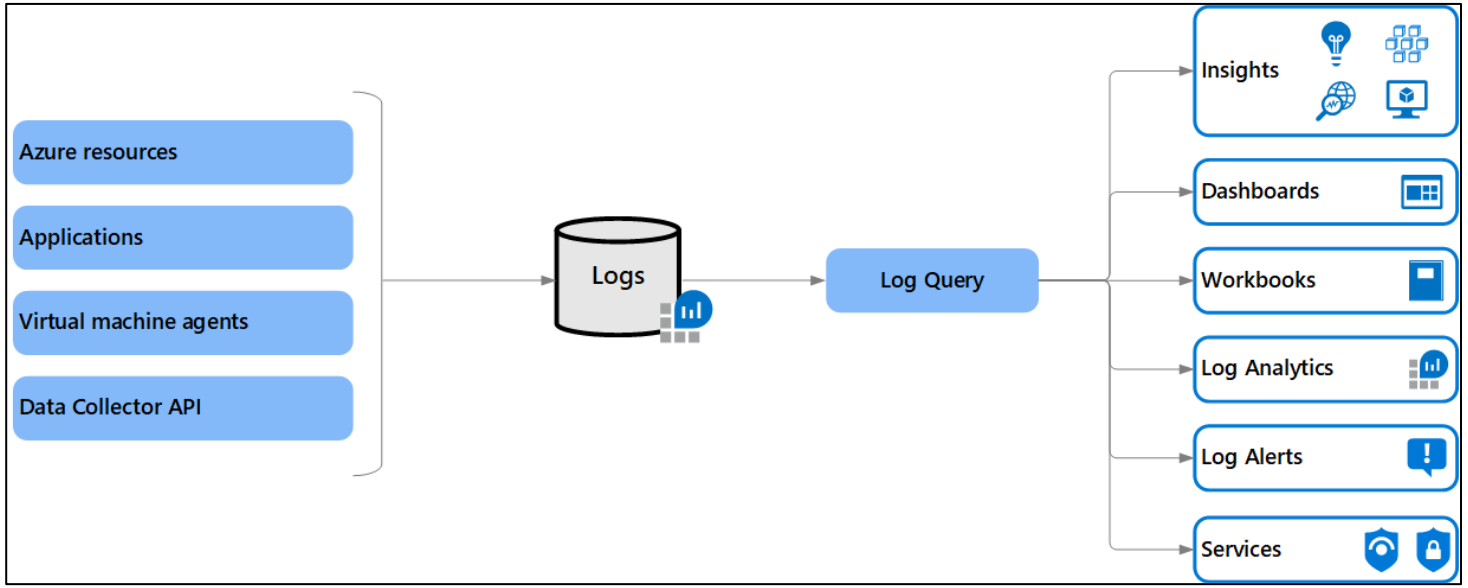
# Agenda

- Azure Monitor Logs
- Log Analytics Workspace
- Data Collection
- Log Data Structure
- Kusto Query Language

# Azure Monitor Logs

- Collect log and performance data
  - From multiple sources

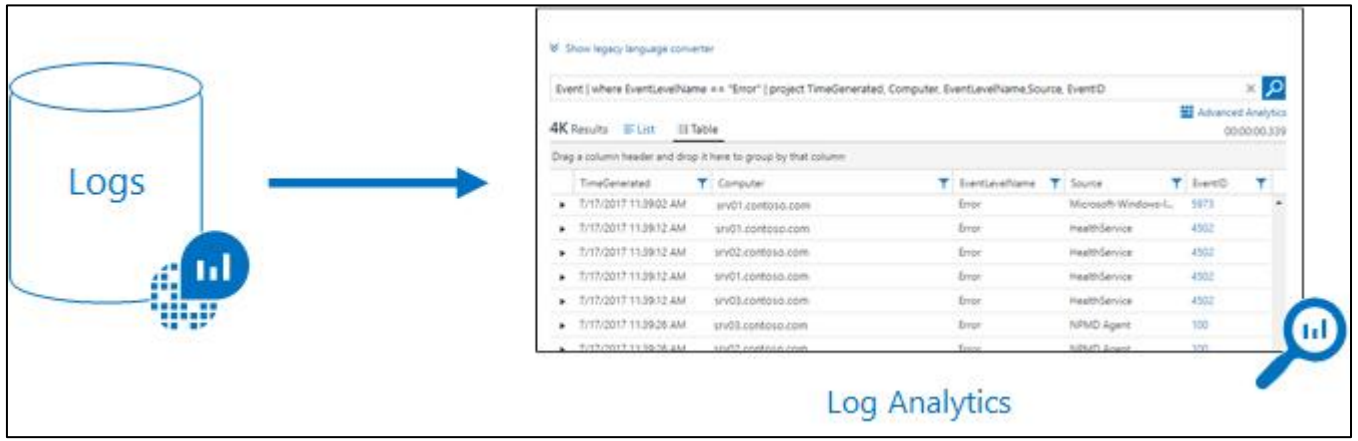
- Act on logged information
  - Analyze
  - Alert
  - Visualize
  - Get insights
  - Retrieve
  - Export



# Log Analytics

- Analyze log data with queries

- Kusto query language
  - Simple log queries
  - Advanced functionality
    - Aggregations
    - Joins
    - Smart analytics



# Data Collection

- Configure sources to send data
  - Azure resources
    - Create diagnostic settings
  - Collect data from VMs
    - Enable VM Insights
  - Collect more events and performance data
    - Configure data sources
- Data collected is stored in Log Analytics workspace

# Workspace

- Container where data is
  - Collected
  - Aggregated
  - Analyzed
  - Presented
- Manage sets of data
  - Collected from your entire IT infrastructure
    - Cloud
    - On-prem

# Workspace

- Provides
  - Geographic location for data storage
  - Data isolation
    - Access rights
  - Scope for configuration settings
    - Pricing tier
    - Retention
    - Data capping

# Pricing

- Data ingestion
  - Capacity reservations
    - Fixed predictable fee

| Capacity | Price |
|---|---|
| 100 GB per day | €233,45 per day |
| 200 GB per day | €438,30 per day |
| 300 GB per day | €643,16 per day |
| 400 GB per day | €838,49 per day |
| 500 GB per day | €1030,24 per day |
| 1000 GB per day | €2024,75 per day |

  - Pay-As-You-Go
    - Billed per GB of data uploaded to service

| | Free units | Price |
|---|---|---|
| Data ingestion | 5 GB per month | €2,761 per GB |

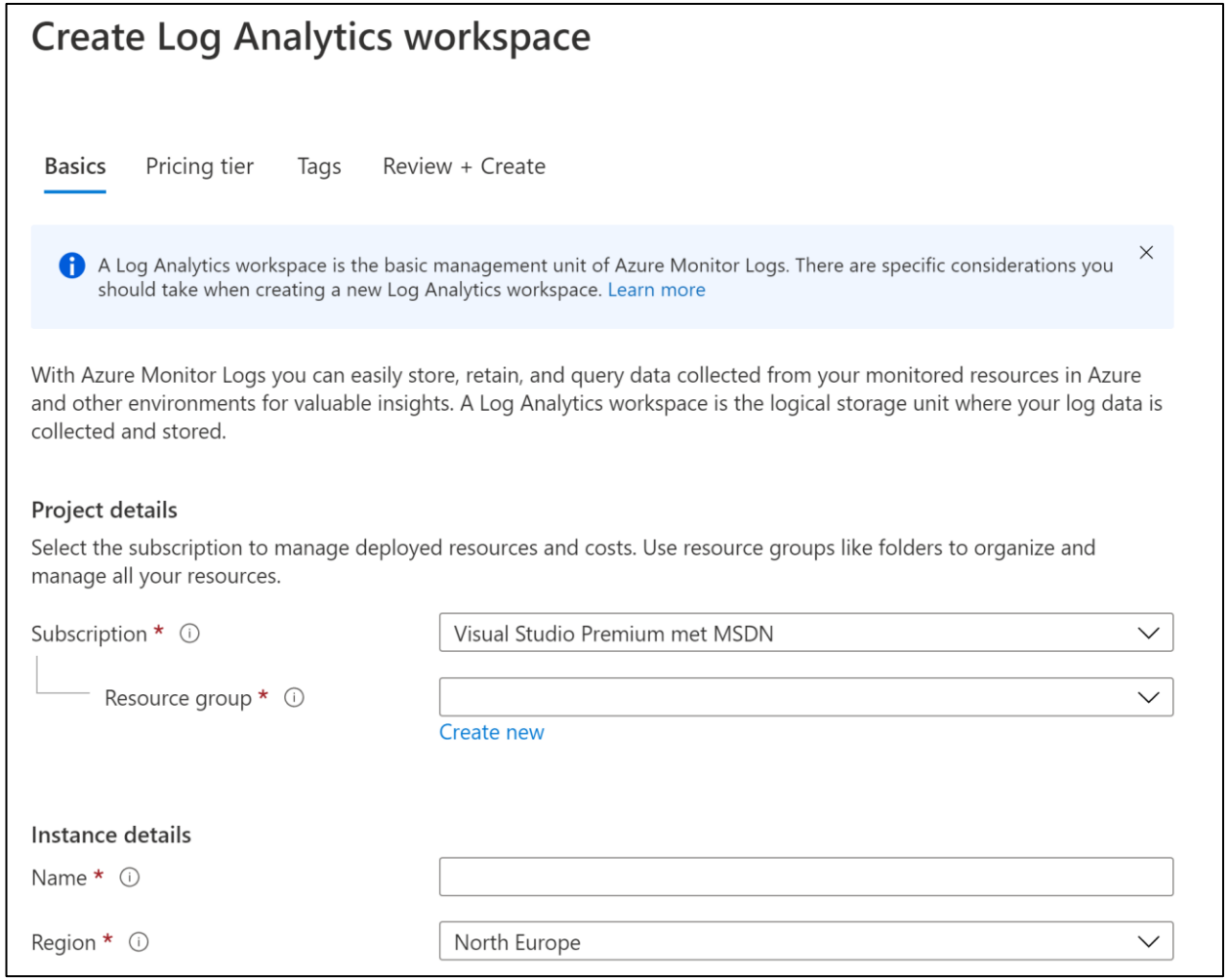
# Pricing

- Data Retention
  - Data is retained for
    - 31 days
      - No extra charge
    - 90 days
      - If Sentinel is enabled
    - 90 days
      - For Application Insights data

| Feature | Free Units | Price |
|---|---|---|
| Data retention | 31 days | €0,121 per GB per month |

# Create Workspace

- Azure Portal
  - Create new resource
  - Log analytics workspace
    - Name
    - Resource group
    - Region



# Data Collection - VMs

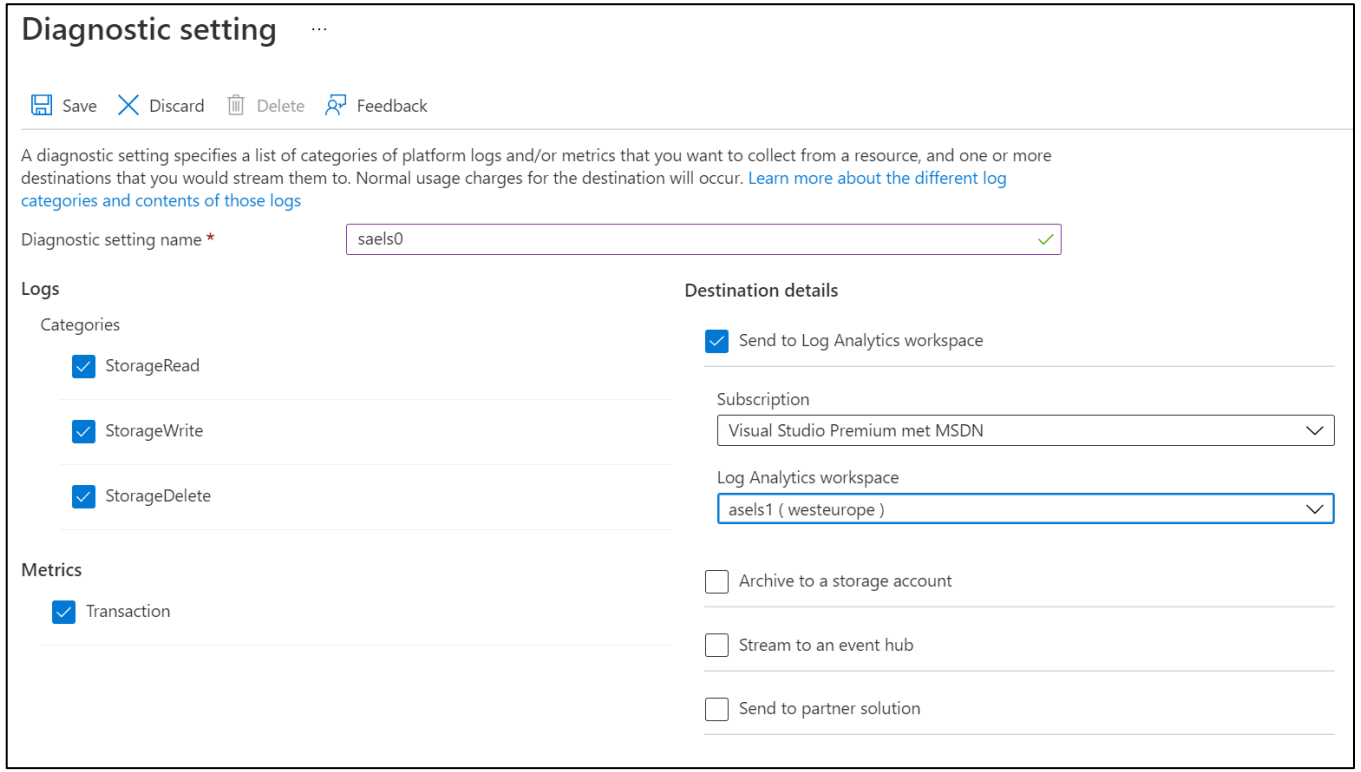
- Agent must be deployed
  - Connected to log analytics workspace
- 2 agents
  - Azure Monitor Agent
  - Log Analytics Agent
    - Legacy

# Azure Monitor Agent

- On-prem VM
  - Install Azure Arc agent
- Create data collection rule set
  - Supported systems
    - Azure VMs
    - Azure Arc enabled VMs
  - Define what to collect
    - Logs
    - Performance counters

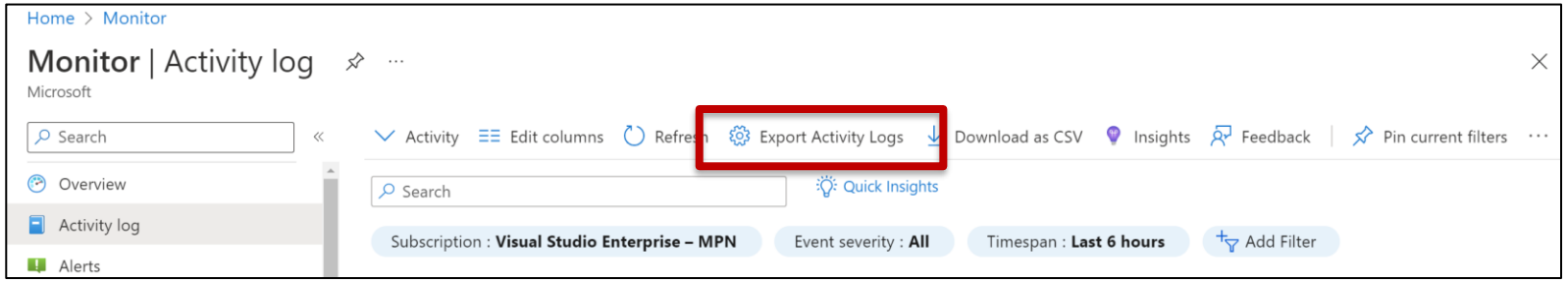
# Data Collection – Azure Resources

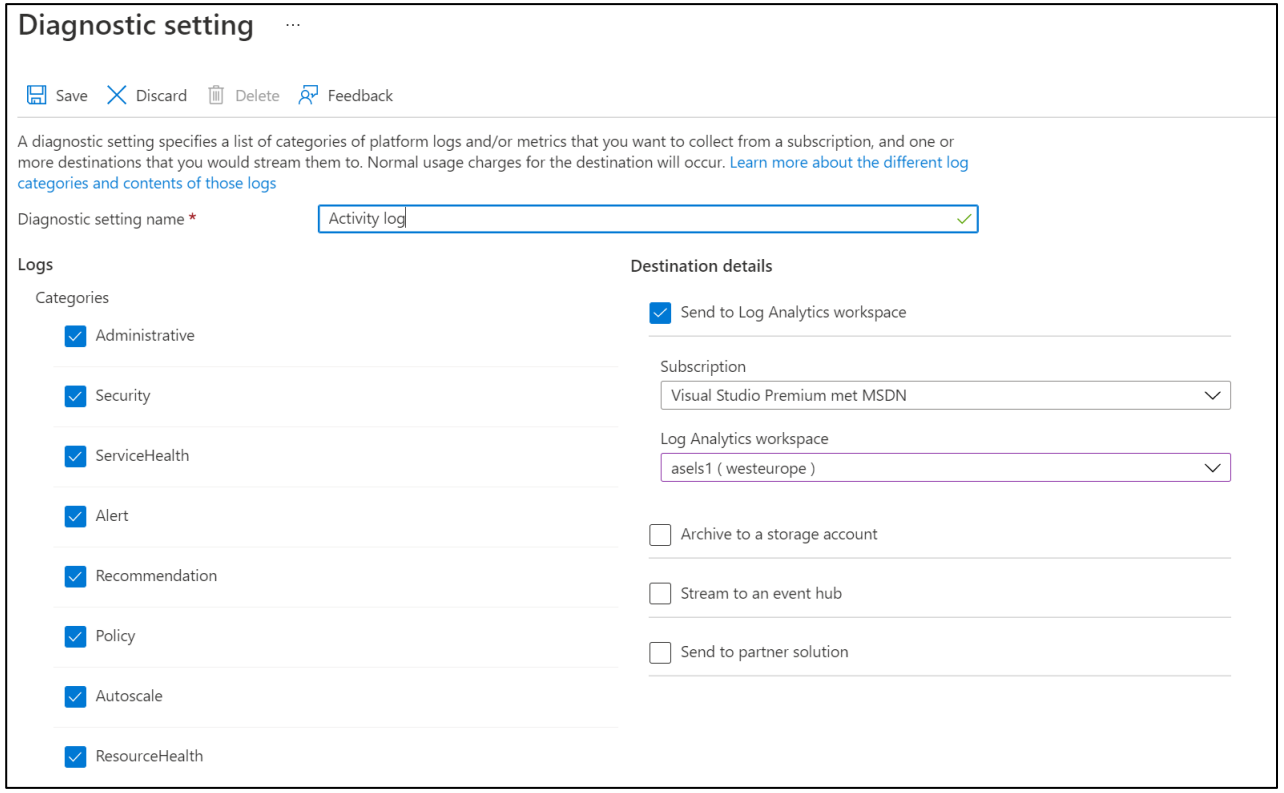
- Go to Azure resource
  - Monitoring – Diagnostic settings
    - Add diagnostic setting
      - Select Log categories
      - Select Metrics
      - Destination details
        - Log Analytics workspace



# Data Collection – Activity Log

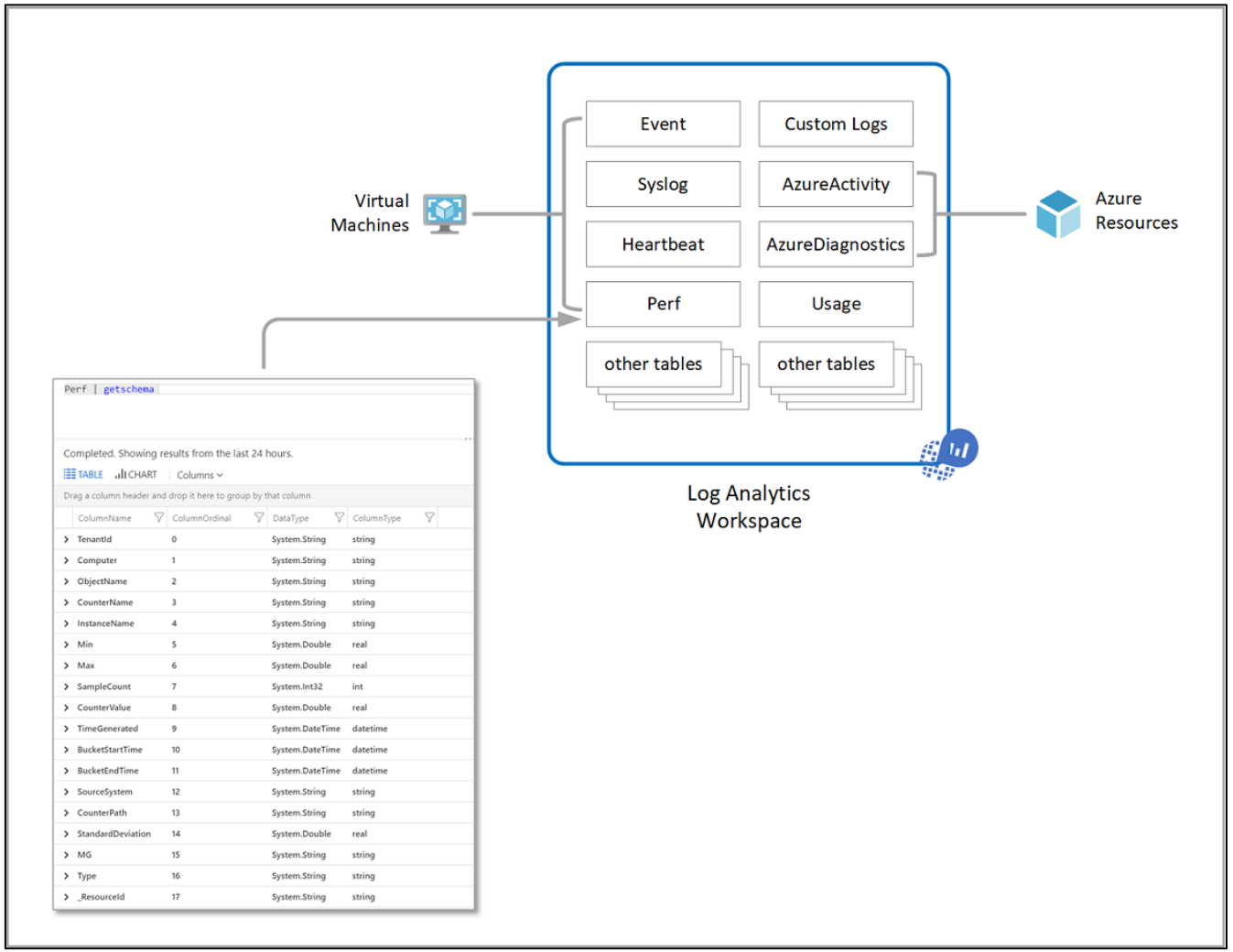
- Azure Monitor
  - Activity log
    - Export Activity Logs



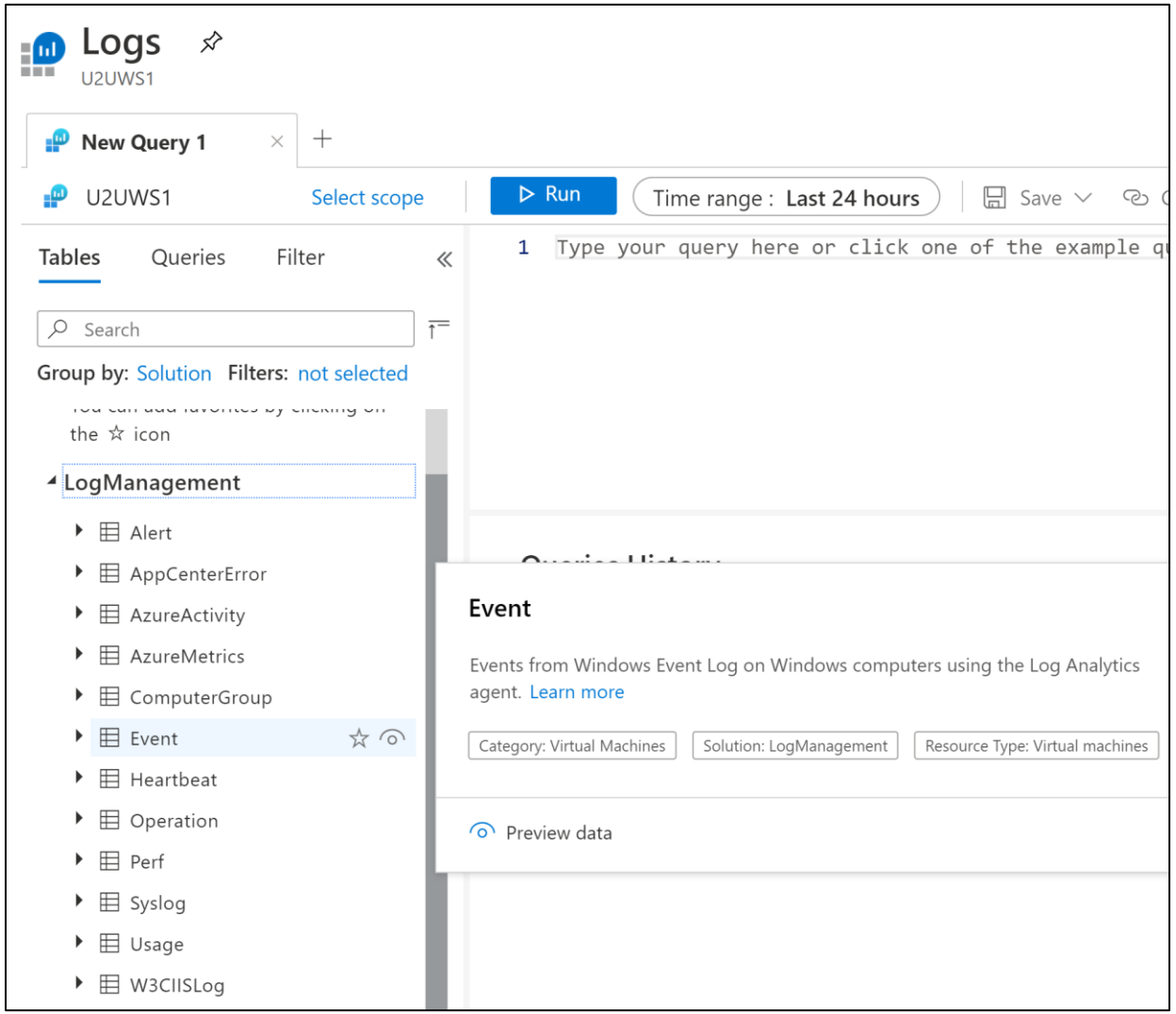
# Data Structure

- Workspace contains tables
  - Organized into columns
    - With multiple rows of data



# Tables

- Data is stored in tables
  - Syslog
  - Heartbeat
  - Event
  - Operation
  - …
- Explore tables
  - Log Analytics workspace
    - Logs
      - Table pane



# Kusto Query Language

- Tool to
  - Explore data and discover patterns
  - Identify anomalies
- Uses schema entities organized in a hierarchy similar to SQL
  - Databases
  - Tables
  - Columns
- Kusto query
  - Read-only request to
    - Process data
    - Return results
  - Easy to read and author
  - Case-sensitive

# Kusto Query Language

- KQL is used with
  - Azure Monitor
  - Azure Data Explorer
- Some differences
  - Operators not supported by Azure Monitor
  - Operators specific to Azure Monitor

- Practice writing KQL statements in a demo environment
  - https://aka.ms/lademo

# Kusto Query Language (KQL)

- Powerful query language
  - Join data from multiple tables
  - Aggregate large sets of data
  - Perform complex operations with minimal code

# Sample Queries

- Retrieve all records from a table
- Retrieve records from table
  - Filter records
  - Summarize
  - Visualize results in chart
- Retrieve data from multiple tables
  - Using a join
  - Analyze combined results

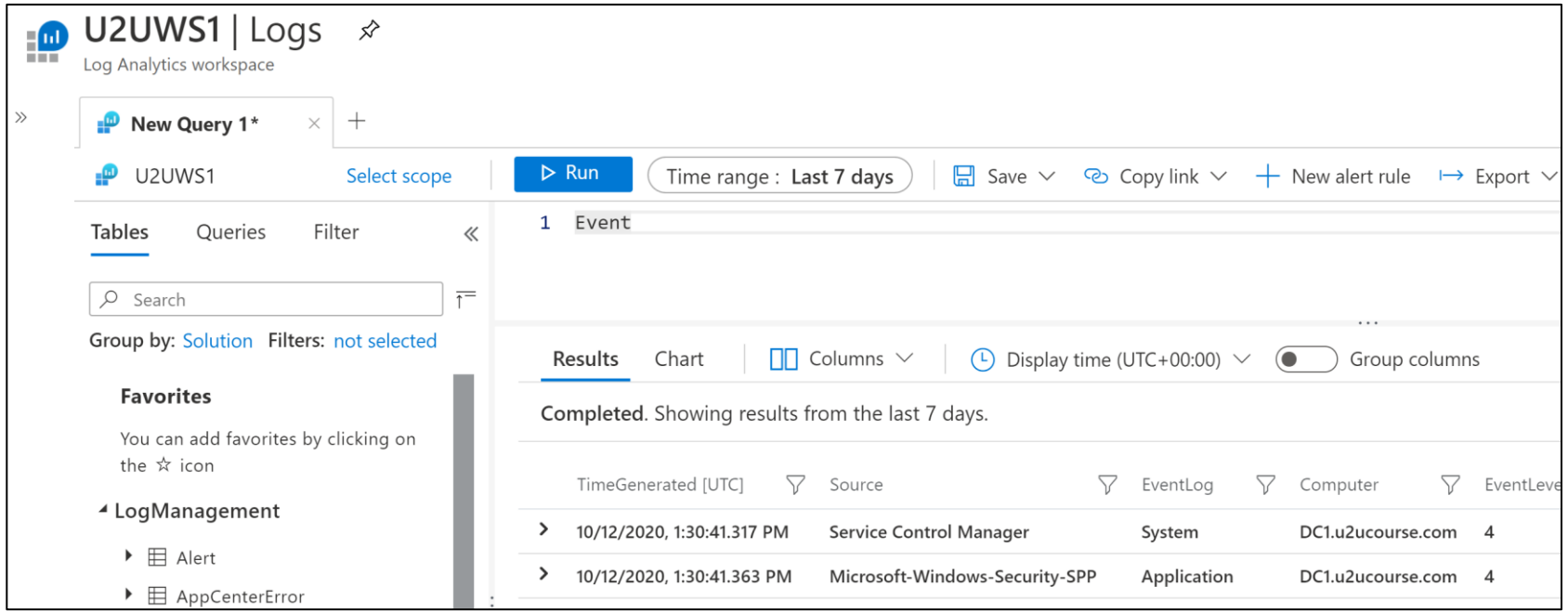
```
Events
```

```
SecurityEvent
| where TimeGenerated > ago(7d)
| where EventID == 4625
| summarize count() by Computer, bin(TimeGenerated, 1h)
| render timechart
```

```
app("ContosoRetailWeb").requests
| summarize count() by bin(timestamp,1hr)
| join kind= inner (Perf
        | summarize avg(CounterValue)
          by bin(TimeGenerated,1hr))
on $left.timestamp == $right.TimeGenerated
```

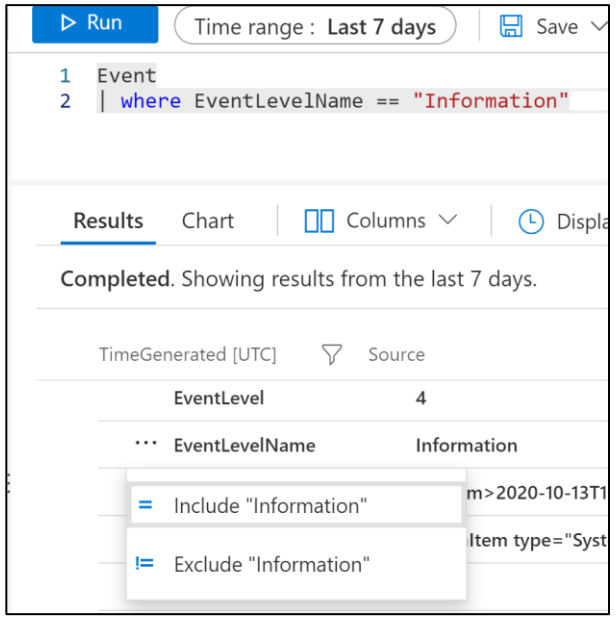
# Run a Query

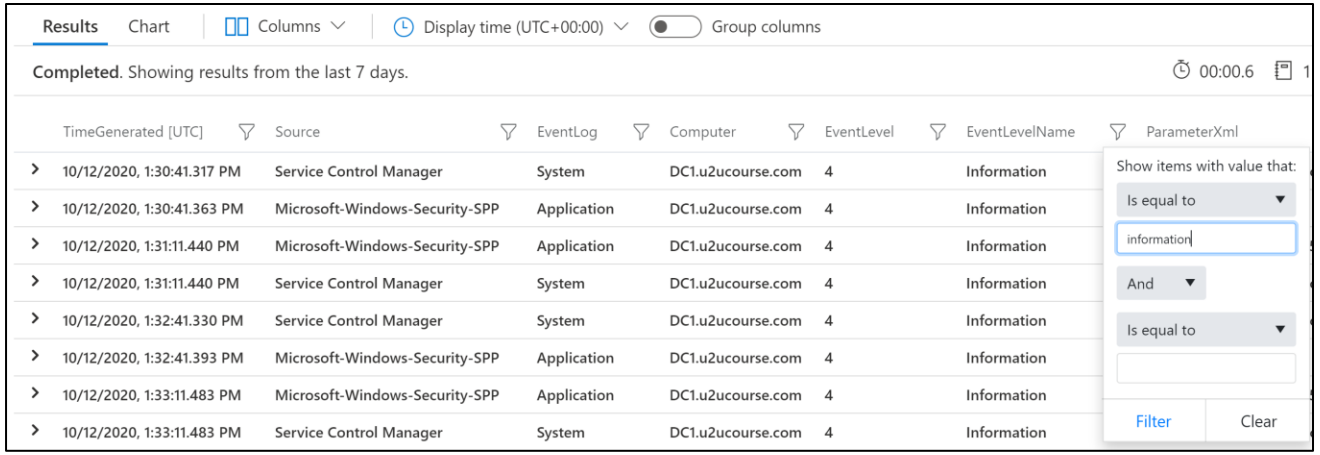
- Place cursor in query window
  - Click Run
  - Shift+Enter
- Use the Time range control
  - Default
    - Last 24 hours



# Filter Results

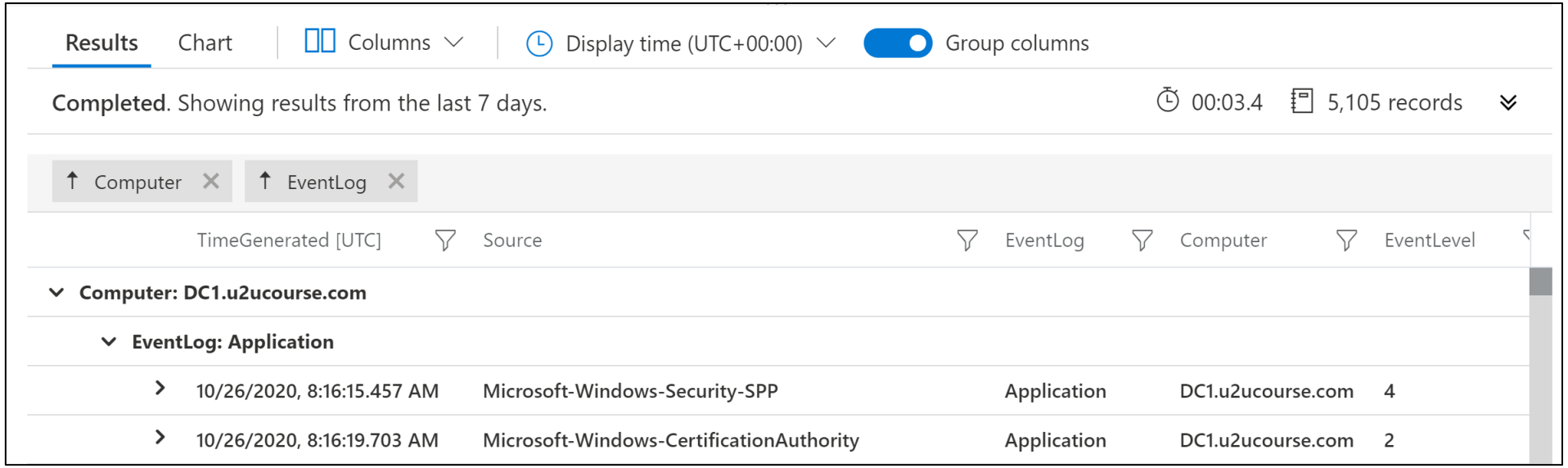
- Restrict table elements
  - From results
    - Select a property
    - Click 3 dots (...)
      - Include/exclude
  - In query
    - Where Propertyname == 'Value'
- Filter results
  - Select Filter icon next to column heading



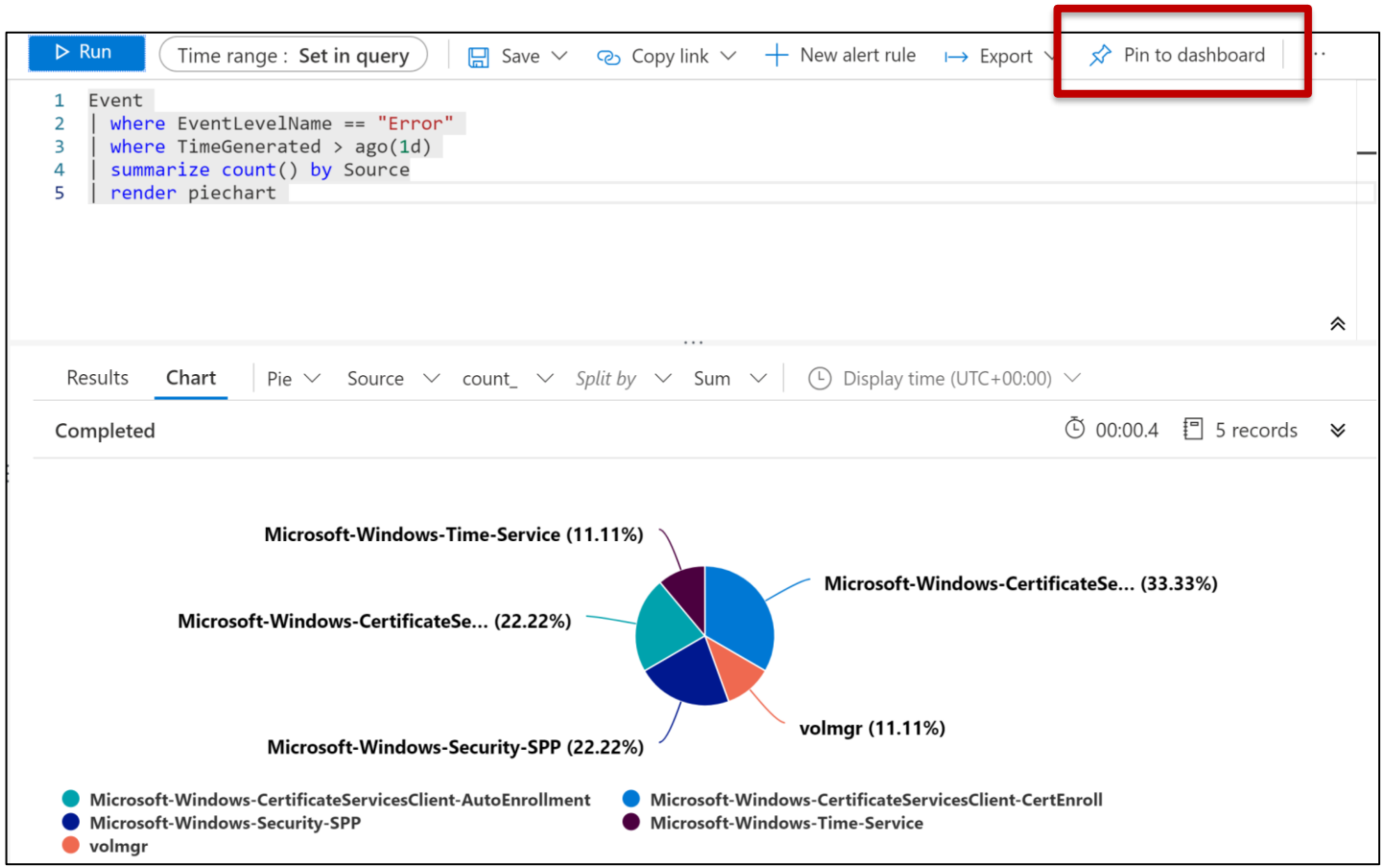
# Sort or Group Columns

- Sort results
  - Click column header
- Group by
  - Drag column header to bar above results table



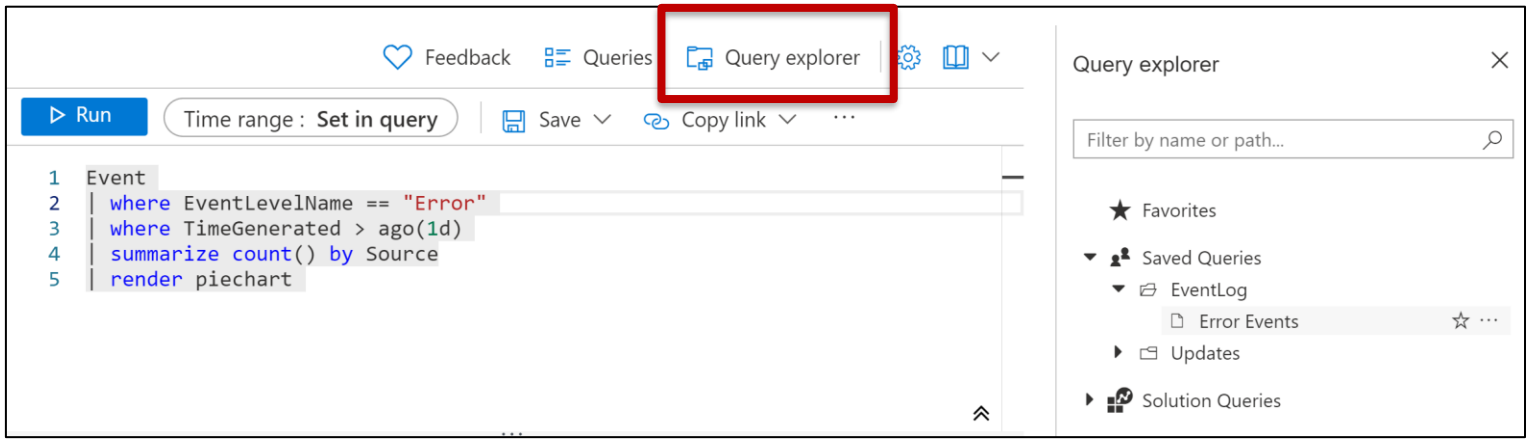
# View and Modify Charts

- Graphic views
  - Chart
  - Stacked Bar
  - Stacked column
  - Pie
- Pin results to dashboard



# Manage Queries

- Save queries
- Load queries
  - Query explorer
  - Mark query as favorite
- Export and share queries
  - Export to
    - .csv
    - PowerBI
  - Share a link to a query



# Write New Query

- Table-based query
  - Find tables in schema pane
  - Defines clear scope
  - Improves query performance

```
Event
| take 10
```

- Search-based query
  - Find records that meet criteria
  - Less structured
  - Slower
  - Search across multiple tables

```
search "Hyper-V"
| take 10
```

```
search in (Event) "Hyper-V"
| take 10
```

# Take - Sort - Top

- Take
  - Return specific number of arbitrary records

```
Event
| take 10
```

- Sort
  - Sort by preferred column
  - Sorts entire table
    - Could return many results
    - Could take long time

```
Event
| sort by Source asc
```

- Top
  - Get latest records
  - Sorts table on server side
    - Returns only top records

```
Event
| top 10 by TimeGenerated
```

# Filter Data

- Where
  - Filter data by a specific condition
  - Limit query results

- Expressions

| Expression | Description | Example |
|---|---|---|
| == | Check equality (case-sensitive) | Level == 8 |
| =~ | Check equality (case-insensitive) | EventSourceName =~ "microsoft" |
| !=, <> | Check inequality | Level != 4 |
| and, or | Combine conditions | Level == 16 or CommandLine != "" |

```
Event
| where EventLog == "Application"
```

```
Event
| where EventLog == "Application" and EventLevel == 1
```

# Specify Time Range

- Time picker
  - Default 24 hours
- Time filter in query
  - Place time filter immediately after table name
  - Time units
    - Days (d)
    - Minutes (m)
    - Seconds (s)

```
Event
| where TimeGenerated > ago(30m)
```

# Select and Compute Columns

- Project
  - Select specific columns to include in results
  - Rename columns
  - Define new columns
  - Variations
    - Project-away
    - Project-keep
    - Project-rename
    - Project-reorder

```
Event
| take 50
| project TimeGenerated, Computer, EventLog, EventLevel
```

```
Event
| take 50
| project TimeGenerated, Computer, LogName=EventLog
```

- Extend
  - Keep all original columns in results
  - Define additional ones
    - Calculated columns

```
Event
| take 50
| extend LogName=EventLog
```

# Aggregate Groups of Rows

- Summarize
  - Identify groups of records
  - Based on columns
  - Apply aggregations
    - Count
      - Return number of results in each group
    - Dcount
      - Returns estimate for distinct values
    - Avg
      - Perform mathematical calculations
    - Max, min
    - Sum

```
Perf
| where TimeGenerated > ago(1h)
| summarize by ObjectName
```

```
Perf
| where TimeGenerated > ago(1h)
| summarize count() by ObjectName
```

```
Perf
| where TimeGenerated > ago(1h)
| summarize count() by ObjectName, CounterName
```

```
Perf
| where TimeGenerated > ago(1h)
| summarize avg(CounterValue) by Computer, CounterName
```

# Summarize by a Time Column

- Group results based on time
  - Summarize by TimeGenerated
    - Creates groups for every millisecond over time range
  - Break range into manageable units
    - Bin
- Example
  - Analyze perf records for free memory on computer
    - Available Mbytes
  - Calculate average value of each 1-hour period over last 7 days

```
Perf
| where TimeGenerated < ago(7d)
| where Computer == "DC1.u2ucourse.com"
| where CounterName == "Available MBytes"
| summarize avg(CounterValue) by bin(TimeGenerated, 1h)
| render timechart
```

# Visualizations

- Render
  - Generate visualization of query results
  - Supported options
    - Areachart
    - Barchart
    - Columnchart
    - Piechart
    - Scatterchart
    - Timechart

```
SecurityEvent
| summarize count() by Account
| render barchart
```

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

# Joins

- Analyze data from multiple tables in same query
  - Merge rows of two datasets
  - Match values of specified columns
- Join flavors
  - Innerunique (default)
    - Inner join with left side deduplication
  - Inner
  - Fullouter
- Join datasets with different key

```
Table1
| join ( Table2 )
on $left.key1 == $right.key2
```

```
Heartbeat
| where OSType == "Windows"
| project ComputerIP, Computer, OSType
| join (
    Event
    | where EventLevelName == "Error"
    | project EventLevelName, Computer, Message
) on Computer
```