# SPL to ES|QL Equivalency Guide

## Overview

This document provides a comprehensive mapping of Splunk Processing Language (SPL) commands and functions to their equivalent Elasticsearch Query Language (ES|QL) commands and functions, ensuring a smooth transition between the two query languages.

---

## Command Equivalency Mappings

### General Processing Commands

**eval**

> **SPL Command:** eval
> *Description (SPL):* Computes new fields based on expressions, conditions, or mathematical calculations.
>
> **ES|QL Equivalent:** EVAL
> *Description (ES|QL):* Used to modify or create fields by applying transformations or calculations.

**Examples:**
SPL Example: `index=users | eval total_price=price*quantity`
ES|QL Equivalent:
FROM users

| EVAL total_price = price * quantity

**where**

> **SPL Command:** WHERE
> **Description (SPL):** Filters events based on a specified condition.
> **ES|QL Equivalent:** WHERE
> **Description (ES|QL):** Filters documents that meet a specified condition.

**Examples:**
**SPL Example:** `index=users | where status="active"`
**ES|QL Equivalent:**

```
FROM users
```

```
| WHERE status = "active"
```

**lookup**

**SPL Command:** lookup
*Description (SPL):* Performs an external lookup to enrich event data.

**ES|QL Equivalent:** ENRICH
*Description (ES|QL):* Adds additional data to records from an external dataset or lookup index.

**Examples:**
SPL Example: `| lookup user_info user_id OUTPUT user_name, email`
ES|QL Equivalent:

```
FROM events
| ENRICH user_info ON user_id
```

**table**

**SPL Command:** table
*Description (SPL):* Keeps only the specified fields.

**ES|QL Equivalent:** KEEP
*Description (ES|QL):* Retains only the specified fields in the output.

**Examples:**
SPL Example: `| table user_id, user_name, email`
ES|QL Equivalent:

```
FROM users
| KEEP user_id, user_name, email
```

**stats**

**SPL Command:** stats
*Description (SPL):* Computes aggregations such as count, sum, average, min, max, etc.

**ES|QL Equivalent:** STATS
*Description (ES|QL):* Performs aggregations like COUNT, SUM, AVG, MIN, MAX over grouped data.

**Examples:**
SPL Example: `index=users | stats count by country`
ES|QL Equivalent:
FROM users

| STATS COUNT(*) BY country

**spath**

**SPL Command:** spath
*Description (SPL):* Extracts values from JSON or structured data formats.

**ES|QL Equivalent:** GROK or DISSECT
*Description (ES|QL):* Extracts values using pattern matching from structured text fields.

**Examples:**
SPL Example: `| spath input=raw_json path=user.name`
ES|QL Equivalent:
FROM logs

| GROK raw_json '%{DATA:user.name}'

**rex**

**SPL Command:** rex
*Description (SPL):* Extracts values using regular expressions.

**ES|QL Equivalent:** GROK or DISSECT
*Description (ES|QL):* Uses regular expressions or pattern-based extraction to retrieve values.

**Examples:**
SPL Example: `| rex field=message "(?<error_code>\d{3})"`
ES|QL Equivalent:
FROM logs

| GROK message '%{NUMBER:error_code}'

**regex**

**SPL Command:** regex
*Description (SPL):* Filters records using regex-based conditions.

**ES|QL Equivalent:** GROK with WHERE/RLIKE
*Description (ES|QL):* Applies regex-based conditions to filter records.

**Examples:**
SPL Example: `| regex message="error|failure"`
ES|QL Equivalent:
FROM logs

| WHERE message RLIKE 'error|failure'

**convert**

**SPL Command:** convert
*Description (SPL):* Converts field types.

**ES|QL Equivalent:** DATE_FORMAT, TO_INTEGER, TO_LONG, etc.
*Description (ES|QL):* Changes data types for fields.

**Examples:**
SPL Example: `| convert num(salary)`
ES|QL Equivalent:
FROM employees

| EVAL salary = TO_INTEGER(salary)

**bin**

**SPL Command:** bin
*Description (SPL):* Rounds timestamps to specific intervals.

**ES|QL Equivalent:** DATE_TRUNC
*Description (ES|QL):* Rounds timestamp values to specified time intervals.

**Examples:**
SPL Example: `| bin _time span=1h`

ES|QL Equivalent:
FROM logs

| EVAL rounded_time = DATE_TRUNC('1h', timestamp)

---

## Function Equivalency Mappings

**sort**

> **SPL Function:** sort
> *Description (SPL):* Sorts the result set based on specified field values.
>
> **ES|QL Equivalent:** SORT
> *Description (ES|QL):* Sorts query results in ascending or descending order.

**Examples:**
SPL Example: `| sort -count`
ES|QL Equivalent:
FROM users

| SORT count DESC

**count**

> **SPL Function:** count
> *Description (SPL):* Returns the total count of records matching the query.
>
> **ES|QL Equivalent:** COUNT(field)
> *Description (ES|QL):* Computes the number of occurrences for a given field.

**Examples:**
SPL Example: `| stats count`
ES|QL Equivalent:
FROM logs

| STATS COUNT(*)

**values**

> **SPL Function:** values
> *Description (SPL):* Returns unique values in a field.

**ES|QL Equivalent:** VALUES(field)
*Description (ES|QL):* Retrieves distinct field values.

**Examples:**
SPL Example: `| stats values(user)`
ES|QL Equivalent:
```
FROM logs

| STATS VALUES(user)
```

**case**

**SPL Function:** case
*Description (SPL):* Evaluates conditions and assigns corresponding values.

**ES|QL Equivalent:** CASE
*Description (ES|QL):* Implements conditional logic to transform or categorize field values.

**Examples:**
SPL Example: `| eval status=case(error=1,"Error", warning=1,"Warning", true(),"OK")`
ES|QL Equivalent:
```
FROM logs

| EVAL status = CASE(error = 1, 'Error', warning = 1, 'Warning', 'OK')
```

**in**

**SPL Function:** in
*Description (SPL):* Checks if a field value exists in a predefined list.

**ES|QL Equivalent:** IN
*Description (ES|QL):* Filters records by verifying whether a field value is in a given set.

**Examples:**
SPL Example: `| where status in ("200", "404")`
ES|QL Equivalent:
```
FROM logs

| WHERE status IN ('200', '404')
```

**by**

**SPL Function:** by
*Description (SPL):* Groups results based on a field.

**ES|QL Equivalent:** BY / WITH
*Description (ES|QL):* Groups records based on a specified field.

**Examples:**
SPL Example: `| stats count by user`
ES|QL Equivalent:
FROM logs

| STATS COUNT(*) BY user

**AS**

**SPL Command:** AS
**Description (SPL):** Renames a field to a new name within the search results.
**ES|QL Equivalent:** RENAME
**Description (ES|QL):** Changes the name of a field to a new name.

**Examples:**
**SPL Example:**

```
index=users | eval total_price=price*quantity | table total_price AS
final_price
```

**ES|QL Equivalent:**

```
FROM users
| EVAL total_price = price * quantity
| RENAME total_price AS final_price
| KEEP final_price
```

**and / or / not**

- **SPL Function:** and, or, not
  *Description (SPL):* Logical conditions in filtering.

- **ES|QL Equivalent:** AND, OR, NOT
  *Description (ES|QL):* Applies logical conditions for filtering.

**Examples:**
SPL Example: `| where status="200" AND response_time>500`
ES|QL Equivalent:
FROM logs

| WHERE status = '200' AND response_time > 500

---

# Example Mappings

**eval Example**

**SPL Example:**

index=web_logs | eval status_type=if(status_code>=400,"error","success")

**ES|QL Equivalent:**

FROM web_logs
| EVAL status_type = CASE WHEN status_code >= 400 THEN "error" ELSE "success" END

**stats Example**

**SPL Example:**

index=users | stats count by country

**ES|QL Equivalent:**

FROM users
| STATS COUNT(*) BY country

**timechart Example**

**SPL Example:**

index=transactions | timechart span=1h sum(amount)

**ES|QL Equivalent:**

FROM transactions
| STATS SUM(amount) BY DATE_TRUNC('1h', timestamp)

---

# Conclusion

This document ensures an easy transition from SPL to ES|QL by mapping common commands and functions. More advanced use cases and optimizations can be explored in Elasticsearch documentation.