| **Command/Operator** | **Syntax** | **Description** |
| --- | --- | --- |
| where | where <condition> | Filters records based on a condition. |
| project | project <column1>, <column2> | Selects specific columns to include in the result. |
| distinct | distinct <column> | Returns unique values of a column. |
| summarize | summarize <aggregation> by <column> | Aggregates data using functions like count(), sum(), etc. |
| count | count() | Returns the number of records. |
| avg | avg(<column>) | Calculates the average of a numeric column. |
| min | min(<column>) | Returns the minimum value of a column. |
| max | max(<column>) | Returns the maximum value of a column. |
| sum | sum(<column>) | Sums the values of a numeric column. |
| join | `<table1> | join <table2> on <column>` |
| bin | bin(<column>, <time\_span>) | Groups data into bins of a specified time span or numeric range. |
| ago | ago(<time\_span>) | Returns records from a specified time ago. |
| datetime | datetime(<YYYY-MM-DD>) | Specifies a specific date. |
| make-series | make-series <aggregation> on <column> from <start> to <end> step <step\_value> | Creates a series of data points over a time range. |
| extend | extend <new\_column> = <expression> | Creates a new column based on an expression. |
| parse | parse <column> with <pattern> | Extracts parts of a string into new columns based on a pattern. |
| mv-expand | mv-expand <column> | Expands arrays or lists into multiple rows. |
| project-rename | project-rename <new\_name> = <old\_name> | Renames a column in the result set. |
| project-away | project-away <column1>, <column2> | Removes specified columns from the result set. |
| order by | order by <column> [asc|desc] | sort the results by column |
| top | `top <number> by <column> | return the top N records |
| take | take <number> | Returns the first N records from the result set. |
| lookup | lookup <table> on <column> | Performs a lookup of values in another table. |
| parsejson | parsejson(<column>) | Parses a JSON string into a dynamic object. |
| mv-apply | mv-apply <column> to (<expression>) | Applies an expression to each element of an array or list. |
| union | union <table1>, <table2> | Combines the results of two or more tables. |
| limit | limit <number> | Limits the number of records returned (synonym for take). |
| sort | `sort by <column> [asc | desc]` |
| search | search <expression> | Searches for a term across multiple columns. |
| has | has "value" | Checks if a column contains a value as a whole word. |
| contains | contains "value" | Checks if a column contains a substring. |
| startswith | startswith("value") | Checks if a string starts with a specified value. |
| endswith | endswith("value") | Checks if a string ends with a specified value. |
| has\_any | has\_any("value1", "value2") | Checks if a column contains any of the specified values. |
| between | between(start, end) | Checks if a value falls within a specified range. |
| in | <column> in ("value1", "value2") | Checks if a column's value is in a set of values. |
| not | not <condition> | Negates a condition. |
| matches regex | matches regex "<pattern>" | Matches a string against a regular expression pattern. |
| project-reorder | project-reorder <column1>, <column2>, ... | Reorders columns in the output. |
| datatable | datatable(ColumnName1:Type1, ColumnName2:Type2, ...) [RowValue1, RowValue2, ...] | Defines a table inline with the query. |
| set | set <variable\_name> = <expression> | Sets a variable to an expression. |
| let | let <variable\_name> = <expression> | Declares and assigns a variable. |
| todynamic | todynamic(<column>) | Converts a JSON string to a dynamic object. |
| tostring | tostring(<column>) | Converts a value to a string. |
| toint | toint(<column>) | Converts a value to an integer. |
| todatetime | todatetime(<column>) | Converts a value to a datetime. |
| to\_double | to\_double(<column>) | Converts a value to a double. |
| to\_guid | to\_guid(<column>) | Converts a value to a GUID. |
| to\_long | to\_long(<column>) | Converts a value to a long integer. |
| to\_decimal | to\_decimal(<column>) | Converts a value to a decimal. |
| trim | trim("char", <column>) | Removes specified characters from the start and end of a string. |
| tolower | tolower(<column>) | Converts a string to lowercase. |
| toupper | toupper(<column>) | Converts a string to uppercase. |
| length | length(<column>) | Returns the length of a string. |
| replace | replace("pattern", "replacement", <column>) | Replaces occurrences of a pattern in a string. |
| startswith\_cs | startswith\_cs("value") | Case-sensitive version of startswith. |
| contains\_cs | contains\_cs("value") | Case-sensitive version of contains. |
| has\_cs | has\_cs("value") | Case-sensitive version of has. |
| hasprefix | hasprefix("value") | Checks if a column starts with a specified prefix. |
| hassuffix | hassuffix("value") | Checks if a column ends with a specified suffix. |
| coalesce | coalesce(<expr1>, <expr2>, ...) | Returns the first non-null expression. |
| iff | iff(<condition>, <true\_value>, <false\_value>) | Returns one of two values based on a condition. |
| case | case(<condition1>, <value1>, <condition2>, <value2>, ...) | Evaluates conditions and returns corresponding values. |
| range | range x from <start> to <end> step <step\_value> | Generates a series of numbers or dates. |
| typeof | typeof(<column>) | Returns the data type of a column. |
| to\_real | to\_real(<column>) | Converts a value to a floating-point number. |
| serialize | serialize <column\_name> = <expression> | Forces evaluation of an expression before serialization. |
| isnull | isnull(<column>) | Checks if a column value is null. |
| isempty | isempty(<column>) | Checks if a column value is empty. |
| isnotempty | isnotempty(<column>) | Checks if a column value is not empty. |