**Shortcuts**

**<enter> //search  
<ctrl>+<space> //suggestions  
<shift>+<enter> //new Line**

**Examples**

***source* | *select* | *where* | *limit* | *order by* | *group* | *'|'external functions***

**source in ('*13151*', '*13158*')  
source in ('*Microsoft Security*', '*Palo Alto*')  
source like '*Microsoft Secu%*'  
source = '*Microsoft Security*'  
sourcetype = '*Security*'  
sourcetag = '*SourceTag*'  
select \* //returns all columns and rows  
select *col1*, *col2* //returns the selected column values  
where '*palo alto*' //performs full-text search  
where *col1* > *20* or (*col2* = '*palo*' and *col3* != '*host*') //multiple conditional statement  
where *col1* between *123* and *456*  
select *col1*, *col2* group  
select *col1*, *col2*, count(\*) group having count(\*) > *20*  
select *col1*, *col2* order by *col1* asc, *col2* desc**

**source**

Used for source name filtering

**source = [*source name or source id*]  
source in ([*source name or source id, ...*])  
sourcetype //filters by source definition code  
sourcetag //filters by source tag  
sourcecategory //filters by source category  
lookup = "\*" //returns lookup data  
geolocation = "\*" //returns geographic data  
alert = "\*" //returns created alerts data**

**Examples**

**source = *'Microsoft Security'*  
source in (*'logsource1'*, *'logsource2'*)  
lookup = "*lookup1*"**

**select**

The column names to be retrieved are specified with the 'select' command

**select [\*|*columns...*|*aggregation functions...*|*as*]**

Examples

**select \* //returns all columns and rows  
select *col1*, *col2* //returns the selected column values  
select *col1*, *col2* as '*newcol2*' //changes col2 name as newcol2**

**where**

Used to filter records

| **Operator** | **Description** | **Usage** |
| --- | --- | --- |
| = | Equal | where *col1* = '*foo*' or *col1* = *123* |
| != | Not Equal | where *col1* != '*foo*' or *col1* != *123* |
| > | Greater Than | where *col1* > *123* |
| < | Less Than | where *col1* < *123* |
| >= | Greater Than or Equal | where *col1* >= *123* |
| <= | Less Than or Equal | where *col1* <= *123* |
| between | Between two values | where *[column]* beetween *[value1]* and *[value2]* |
| in | If it is in the list | where *[column]* int (*[value1]*, *[value2]*, ...) |
| notin | If it is not in the list | where *[column]* notin (*[value1]*, *[value2]*, ...) |
| like | Like operator | where *[column]* like *'[value]%'* **//%value, value%, val%ue can be used.** |
| notlike | Not like operator | where *[column]* notlike *'[value]%'* |

Examples

**where '*palo alto*' //performs full-text search  
where *col1* > *20* or (*col2* = '*palo*' and *col3* != '*host*') //multiple conditional statement  
where *col1* in ('*foo*', '*bar*') //col1 with values foo and bar  
where *col1* notin (*'foo'*, *'bar'*)  
where *col1* between *123* and *456*  
where *col1* like *'host%'* //starting with 'host'**

**group**

Groups the result set by one or more columns

If used with aggregate functions, 'group' keyword can be omitted. 'having' can be used to filter the result after aggregation functions

**select [*group columns...*, *aggregation functions...*] group [*group columns*] having [*group conditional expression*] top [*number*]**

Aggregation Functions

| **Name** | **Description** | **Usage** |
| --- | --- | --- |
| count | Calculates row count | select count([*column*|\*]) |
| sum | Calculates the sum of the numeric values in the column | select sum(*column*) |
| avg | Calculates the average of the numeric values in the column | select avg(*column*) |
| min | Get the lowest value for the selected column | select min(*column*) |
| max | Get the highest value for the selected column | select max(*column*) |

Examples

**select count(\*)  
select count(\*), max(*col1*), avg(*col2*), sum(*col3*), min(*col4*)  
select *col1*, count(\*)  
select *col1*, *col2*, count(\*) group *col1*, *col2*  
select *col1*, *col2*, count(\*) group having count(\*) > *20*  
select *col1*, *col2*, count(\*), sum(*col1*) group having count(\*) > *20* and sum(*col1*) < *4000* top *10*  
select *col1*, *col2*, count(\*), sum(*col1*) having count(\*) > *20* and sum(*col1*) < *4000***

**order by**

Used to sort columns in ascending or descending order

**select [*columns...*] order by [*columns...*] [asc|desc]**

Examples

**select *col1*, *col2* order by *col1* desc  
select *col1*, *col2* order by *col1* asc, *col2* desc  
select *col1*, *col2* order by *col1* asc, *col2* desc  
select *col1*, *count(\*)* order by *count(\*)* desc //use on a aggregation function  
select *col1*, *ts*, *count(\*)* order by *ts* desc**

**limit**

Limits the result set

**limit [*number*]**

Examples

**select \* limit *10***