manticoresearch 3 Cheatsheets

SELECT

Simple select
```
SELECT * FROM myindex
```

MATCH('full-text query expression')
```
SELECT * FROM myindex WHERE MATCH('find me')
```

GROUP BY
```
SELECT * FROM myindex WHERE MATCH('...') GROUP BY col1;
```

GROUP n BY
return n rows from each group
```
SELECT * FROM myindex WHERE MATCH('...') GROUP 5 BY col1;
```

WITHIN GROUP ORDER BY
non-standard SQL, allows ordering rows inside a group
```
SELECT * FROM myindex WHERE MATCH('...') GROUP by col1 WITHIN GROUP BY ORDER by col2 DESC order by col3 ASC
```

HAVING
```
SELECT * FROM myindex WHERE MATCH('...') GROUP BY col1 HAVING col2>10
```

ORDER BY
```
SELECT * FROM myindex WHERE MATCH('...') ORDER BY WEIGHT() DESC, col1 ASC
```

LIMIT
```
SELECT * FROM myindex LIMIT 10,20
```

FACET {expr_list} [BY {expr_list}] [ORDER BY {expr | FACET()} {ASC | DESC}] [LIMIT [offset,] count]
returns additional result sets by using grouping on the matched and filtered result set.
```
SELECT * FROM test FACET brand_name BY brand_id ORDER BY brand_name ASC FACET property;
```

OPTION
fine tunning query settings
```
SELECT * FROM myindex LIMIT 10,20 OPTION opt1=value1,opt2=value2
```

Query additional information and profiling

SHOW META [LIKE pattern]
shows additional meta-information about the latest query

SHOW WARNINGS
retrieve the warning produced by the latest query

SHOW PLAN [OPTION format=dot]
displays the execution plan of the previous SELECT statement (requires `SET profiling=1`)

SHOW PROFILE
detailed execution profile of the previous SQL statement (requires `SET profiling=1`)

EXPLAIN QUERY index query [OPTION format=dot]
displays execution tree of the provided full-text query without performing actual search over the index

Replication commands

CREATE CLUSTER name ['/path/' as path [,'ip1:port1,ip2:port2' as nodes]]
Initialize a cluster

JOIN CLUSTER name AT 'ip1:port1'
Join a cluster by connecting to one of it's nodes

JOIN CLUSTER name 'ip1:port1,ip2:port2; as nodes,'/path/to' as path
Join a cluster by explicit define cluster nodes and custom path to metadata files

ALTER CLUSTER name UPDATE nodes
Issue updating the list of active nodes across the cluster

DELETE CLUSTER name
Delete a cluster (all nodes are released)

ALTER CLUSTER name ADD|DROP index
Add or remove an index from a cluster

Index manipulation

CREATE TABLE [IF NOT EXISTS] name [(create_definition,...)] [table_options]
Create an index (RT,PQ or distributed) [RT mode only]

CREATE TABLE [IF NOT EXISTS] name LIKE old_name
Create an index using schema and settings of an existing index [RT mode only]<

DROP TABLE name
Drop an existing index [RT mode only]<

ALTER TABLE index {ADD|DROP} COLUMN column_name [{INTEGER|INT|BIGINT|FLOAT|BOOL|MULTI|MULTI64|JSON|STRING}]
Change index structure

ALTER TABLE myrtindex rt_mem_limit='xxx'
Change rt_mem_limit on-the-fly. RT mode only.

ATTACH INDEX diskindex TO RTINDEX rtindex
move data from a regular disk index to a RT index

ALTER TABLE delta KILLLIST_TARGET='main'
Change killlist_target of an index

RELOAD INDEX idx [FROM '/path/to/index_files']
Rotate specific index

RELOAD INDEXES
Rotate indexes

TRUNCATE TABLE rtindex [WITH RECONFIGURE]
Truncate RealTime data

OPTIMIZE INDEX index_name [sync=1,[cuttoff=0]]
Optimize a RealTime index

{DESC | DESCRIBE} index [LIKE pattern]
Lists index columns and their associated types

{DESC | DESCRIBE} index TABLE
Lists expected document schema (PQ only)

FLUSH RTINDEX rtindex
Flushes RT index RAM chunk contents to disk

FLUSH RAMCHUNK rtindex
Create new disk chunk from RT index RAM chunk

SHOW CREATE TABLE name
Prints the CREATE TABLE statement that creates the named table

IMPORT TABLE table_name FROM 'path'
Imports a peviously created RT index into a daemon running in RT mode

Data manipulation

INSERT
```
INSERT INTO myindex(id,field1,field2,col1,col2) VALUES(1,'title','content',10,100);
```

REPLACE
```
REPLACE INTO myindex(id,fiel d1,field2,col1,col2) VALUES (1,'title','content',10,100);
```

UPDATE
```
UPDATE myindex SET col2=200 WHERE id=1;
```

DELETE
```
DELETE FROM myindex WHERE id=1;
```

Special statements

CALL PQ(data, index[, opt_value AS opt_name[, ...]])
performs percolate query

CALL {SUGGEST|QSUGGEST}(word, index [,options])
performs word correction

CALL KEYWORDS(text, index [, options])
tokenization tester

CALL SNIPPETS(data, index, query[, opt_value AS opt_name[, ...]])
highlighting tool

DBA

SHOW STATUS [LIKE pattern]
Display daemon performance counters

SHOW AGENT ['agent'|'index'] STATUS [LIKE pattern]
Displays the statistic of remote agents or distributed index

SHOW INDEX index_name STATUS [LIKE pattern]
Displays various per-index statistics

SHOW INDEX index_name[.N | CHUNK N] SETTINGS
Displays per-index settings in a sphinx.conf compliant file format. It can be also specified a particular chunk number in case of a RT index.

SHOW TABLES [LIKE pattern]
Display all active indexes

SET [GLOBAL] server_variable_name = value
SET a global variable

SHOW THREADS [OPTION columns=width|format=sphinxql]
List active client threads

SHOW [{GLOBAL | SESSION}] VARIABLES [WHERE variable_name='xxx']
Display global variables

FLUSH HOSTNAMES
Renew IPs associates to agent host names

FLUSH LOGS
Initiate reopen of log files

FLUSH ATTRIBUTES
Flushes all in-memory attribute updates

SHOW PLUGINS
Displays all the loaded plugins and UDFs

CREATE FUNCTION udf_name RETURNS {INT | INTEGER | BIGINT | FLOAT | STRING} SONAME 'udf_lib_file'
Installs a user-defined function (UDF) with the given name and type from the given library file. The library file must reside in a trusted plugin_dir directory.

CREATE PLUGIN plugin_name TYPE 'plugin_type' SONAME 'plugin_library'
Loads the given library (if it is not loaded yet) and loads the specified plugin from it.

DROP FUNCTION udf_name
DROP FUNCTION statement deinstalls a user-defined function (UDF) with the given name.

DROP PLUGIN plugin_name TYPE 'plugin_type'
Markes the specified plugin for unloading.

Special SELECT functions

GEODIST(lat1, lon1, lat2, lon2, { option=value, ... })
Calculate geodistance between 2 sets of coordinates

SNIPPET(('documents'), 'my.query', 'limit=100')
wrapper for snippets functionality, similar with CALL SNIPPETS statement

HIGHLIGHT({options}, 'field1,field2', 'alternative query')
Highlighting using stored fields as source.

EXIST ("attr-name", default-value)
replaces non-existent columns with default values

WEIGHT()
returns relevancy score. Note that relevancy score because of it's nature is relative to the query.

GROUP_CONCAT()
produces a comma-separated list of the attribute values of all documents in the group

REMOVE_REPEATS (result_set, column, offset, limit)
removes repeated adjusted rows with the same 'column' value.

ZONESPANLIST()
Returns pairs of matched zone spans in case ZONESPAN match operator is used.

SELECT * FROM (SELECT … ORDER BY cond1 LIMIT X) ORDER BY cond2 LIMIT Y
Manticore allows limited subselects. The outer

Cheatsheets v7 - up to Manticore Search 3.5.4
© 2017-2021 Manticore Software Ltd.
https://github.com/manticoresoftware/manticoresearch

Interested on getting help?

Check our support plans at
https://manticoresearch.com/services/

contact@manticoresearch.com

manticoresearch 3 Cheatsheets

Relevancy rankers (for OPTION ranker=xyz)

proximity_bm25
sum(lcs*user_weight)1000+bm25

bm25
bm25

none
1

wordcount
sum(hit_count*user_weight)

proximity
sum(lcs*user_weight)

matchany
sum((word_count+(lcs-1)max_lcs)user_weight)

fieldmask
field_mask

sph04
sum((4*lcs+2(min_hit_pos==1)+exact_hit)*user_weight)*1000+bm25

expr
user custom ranking formula

export
same as expr, but calculates all field level factors available and outputs with RANKFACTORS()

Date and time functions

DAY()
Returns the integer day of month (in 1..31 range) from a timestamp argument, according to the current timezone..

MONTH()
Returns the integer month (in 1..12 range) from a timestamp argument, according to the current timezone..

NOW()
Returns the current timestamp as an INTEGER..

YEAR()
Returns the integer year (in 1969..2038 range) from a timestamp argument, according to the current timezone.

YEARMONTH()
Returns the integer year and month code (in 196912..203801 range) from a timestamp argument, according to the current timezone.

YEARMONTHDAY()
Returns the integer year, month, and date code (in 19691231..20380119 range) from a timestamp argument, according to the current timezone.

SECOND()
Returns the integer second (in 0..59 range) from a timestamp argument, according to the current timezone.

MINUTE()
Returns the integer minute (in 0..59 range) from a timestamp argument, according to the current timezone.

HOUR()
Returns the integer hour (in 0..23 range) from a timestamp argument, according to the current timezone.

Full-text search operators

operator AND
always implicit

Hello World

operator OR
has higher precende than AND

Hello | World

operator NOT
Hello -World

operator MAYBE
similar to OR, but doesn't return only right subtree expression

Hello MAYBE World

field search operator
limit subsequent search to a given field

@title hello @body world

multiple-field search operator
@(title,body) hello world

ignore field search operator
@!title hello world

ignore non-existing field name(specify at the beginning of the query)
@@relaxed @nosuchfield my query

all-field search operator (implicit)
@* hello

phrase search operator
"hello world"

proximity search operator
"hello world"~10

quorum matching operator
"the world is a wonderful place"/3

strict order operator
aaa << bbb << ccc

exact form modifier
raining =cats and =dogs

field-start and field-end modifier
^hello world$

keyword IDF boost modifier
boosted^1.234 boostedfieldend$^1.234

NEAR, generalized proximity operator:
hello NEAR/3 world NEAR/4 "my test"

NOTNEAR, negative assertion operator
Church NOTNEAR/3 street

SENTENCE operator:
all SENTENCE words SENTENCE "in one sentence"

PARAGRAPH operator
"Bill Gates" PARAGRAPH "Steve Jobs"

ZONE limit operator
ZONE:(h3,h4) only in the titles

ZONESPAN limit operator
ZONESPAN:(h2 only in a (single) title)

Comparison functions

IF()
It takes 3 arguments, check whether the 1st argument is equal to 0.0, returns the 2nd argument if it is not zero, or the 3rd one when it is. Note that unlike comparison operators, IF() does not use a threshold

IN(expr,val1,val2,...)
takes 2 or more arguments, and returns 1 if 1st argument (expr) is equal to any of the other arguments (val1..valN), or 0 otherwise.

INTERVAL(expr,point1,point2,point3,...)
takes 2 or more arguments, and returns the index of the argument that is less than the first argumen

Numeric functions

ABS()
Returns the absolute value of the argument.

BITDOT()
BITDOT(mask, w0, w1, ...) returns the sum of products of an each bit of a mask multiplied with its weight.

CEIL()
Returns the smallest integer value greater or equal to the argument.

CONTAINS(polygon, x, y)
checks whether the (x,y) point is within the given polygon, and returns 1 if true, or 0 if false. The polygon has to be specified using either the POLY2D() function or the GEOPOLY2D() function.

COS()
Returns the cosine of the argument.

DOUBLE()
Forcibly promotes given argument to floating point type.

EXP()
Returns the exponent of the argument

FIBONACCI(N)
Returns the N-th Fibonacci number, where N is the integer argument.

FLOOR()
Returns the largest integer value lesser or equal to the argument.

GEOPOLY2D(x1,y1,x2,y2,x3,y3...)
roduces a polygon to be used with the CONTAINS() function. This function takes into account the Earth's curvature by tessellating the polygon into smaller ones.The function expects coordinates to be in degrees

IDIV()
Returns the result of an integer division of the first argument by the second argument.

LN()
Returns the natural logarithm of the argument

LOG10()
Returns the common logarithm of the argument .

LOG2()
Returns the binary logarithm of the argument.

MAX()
Returns the bigger of two arguments..

MIN()
Returns the smaller of two arguments

POLY2D(x1,y1,x2,y2,x3,y3...)
produces a polygon to be used with the CONTAINS() function. .

POW()
Returns the first argument raised to the power of the second argument.

SIN()
Returns the sine of the argument.

SQRT()
Returns the square root of the argument..

UINT()
Forcibly reinterprets given argument to 64-bit unsigned type..

String functions

CONCAT()
concatenates two or more strings into one. Non-string arguments must be explicitly converted to string using TO_STRING() function

SUBSTRING_INDEX(string,delimiter,number)
returns a substring of a string before a specified number of delimiter occurs

REGEX(attr,expr)
perform regular expression matching on attribute value

Cheatsheets v7 - up to Manticore Search 3.5.4
© 2017-2021 Manticore Software Ltd.
https://github.com/manticoresoftware/manticoresearch

Interested on getting help?

Check our support plans at
https://manticoresearch.com/services/

contact@manticoresearch.com

manticoresearch 3 Cheatsheets

CALL PQ options

docs
provide per query documents matched at result set(default 0 - disabled)

docs_id
use a property from JSON doc to be used as document id(default disabled, only to be used with docs=1)

docs_json
input documents are raw strings (0) or JSON objects (1 - default)

```
CALL PQ ('index_name', ('multiple documents',
'go this way'), 0 as docs_json );

CALL PQ ('pq', ('{"title":"header text",
"body":"post context", "timestamp":11
}','{"title":"short post", "counter":7 }') );
```

mode
how to distribute docs about members of a dist index. Possible values: 'sparsed' (default) and 'sharded'

query
provide all query fields stored, such as query, tags, filters (default 0 - disable)

skip_bad_json
in case of bad json document skip it and continue (1) or end with error (0,default)

skip_empty
in case of empty json document silent ignore and continue (1) or treat depending on skip_bad_json (0, default)

shift
a number to be added to doc ids if no 'docs_id' defined.Only for dist. setups

verbose
provide extended info on matching at SHOW META (default 0 - disabled)

CALL SUGGEST options

limit
returned N top matches, default is 5

max_edits
keep only dictionary words which Levenshtein distance is less or equal, default is 4

result_stats
provide Levenshtein distance and document count of the found words, default is 1 (enabled)

delta_len
keep only dictionary words whose length difference is less, default is 3

max_matches
number of matches to keep, default is 25

reject
defaults to 4; rejected words are matches that are not better than those already in the match queue

result_line
alternate mode to display the data by returning all suggests, distances and docs each per one row, default is 0

non_char
do not skip dictionary words with non alphabet symbols, default is 0 (skip such words)

CALL KEYWORDS options

stats
show statistics of keywords, default is 0

fold_wildcards
fold wildcards, default is 1

fold_lemmas
fold morphological lemmas, default is 0

fold_blended
fold blended words, default is 0

expansion_limit
override expansion_limit defined in configuration, default is 0 (use value from configuration)

sort_mode
sort output by 'docs' or 'hits', default no sorting

Highlighting options

before_match
A string to insert before a keyword match. A %PASSAGE_ID% macro can be used in this string.

after_match
A string to insert after a keyword match.

chunk_separator
A string to insert between snippet chunks (passages). Default is ...

field_separator
A string to insert between fields. Default is |

limit
Maximum snippet size, in symbols (codepoints). Integer, default is 256.

around
How much words to pick around each matching keywords block. Integer, default is 5.

exact_phrase
Whether to highlight exact query phrase matches only instead of individual keywords. Boolean, default is false.

use_boundaries
Whether to additionally break passages by phrase boundary characters. Boolean, default is false.

weight_order
Whether to sort the extracted passages in order of relevance or in order of appearance in the document. Boolean, default is false.

query_mode
Whether to handle $words as a query in extended syntax, or as a bag of words (default behavior).

force_all_words
Ignores the snippet length limit until it includes all the keywords. Boolean, default is false.

limit_passages
Limits the maximum number of passages that can be included into the snippet. Integer, default is 0 (no limit).

limit_words
Limits the maximum number of words that can be included into the snippet.

start_passage_id
Specifies the starting value of %PASSAGE_ID% macro (that gets detected and expanded in before_match, after_match strings). Integer, default is 1

load_files
Whether to handle $docs as data to extract snippets from (default behavior), or to treat it as file names, and load data from specified files on the server side.

load_files_scattered
For distributed snippets generation with remote agents and load_files. Whenever to be be sure that all snippets are actually created.Boolean, default is false.

html_strip_mode
HTML stripping mode setting. Defaults to index, which means that index settings will be used. The other values are none,strip and retain.

allow_empty
Allows empty string to be returned as highlighting result when a snippet could not be generated.By default, the beginning of original text would be returned instead of an empty string. Boolean, default is false.

passage_boundary
Ensures that passages do not cross a sentence, paragraph, or zone boundary.String, allowed values are sentence, paragraph, and zone.

emit_zones
Emits an HTML tag with an enclosing zone name before each passage. Boolean, default is false.

force_passages
Whether to generate passages for snippet even if limits allow to highlight whole text. Boolean, default is false.

limits_per_field
Whether limit options should apply to field or whole document. Boolean, default is true.

Type conversion functions

BIGINT()
Forcibly promotes the integer argument to 64-bit type, and does nothing on floating point argument.

INTEGER()
Forcibly promotes given argument to 64-bit signed type

Miscellaneous functions

ALL()
`ALL(cond FOR var IN json.array)` - applies to JSON arrays and returns 1 if condition is true for all elements in array and 0 otherwise.

ALL(mva) - special constructor for multi value attributes. When used in conjunction with comparison operators it returns 1 if all values compared are found among the MVA values.

ANY()
`ANY(cond FOR var IN json.array)` - works similar to ALL() except for it returns 1 if condition is true for any element in array.

ANY(mva) is a special constructor for multi value attributes. When used in conjunction with comparison operators it returns 1 if any of the values compared are found among the MVA values.

ATAN2()
Returns the arctangent function of two arguments, expressed in radians.

CRC32()
Returns the CRC32 value of a string argument

GREATEST()
takes JSON array or MVA as the argument, and returns the greatest value in that array.

INDEXOF(cond FOR var IN json.array)
iterates through all elements in array and returns index of first element for which 'cond' is true and -1 if 'cond' is false for every element in array.

LEAST()
takes JSON array or MVA as the argument, and returns the least value in that array.

LENGTH()
LENGTH(attr_mva) function returns amount of elements in MVA set.

LENGTH(attr_json) returns length of a field in JSON. Return value depends on type of a field.

MIN_TOP_SORTVAL()
Returns sort key value of the worst found element in the current top-N matches if sort key is float and 0 otherwise.

MIN_TOP_WEIGHT()
Returns weight of the worst found element in the current top-N matches.

PACKEDFACTORS()
can be used in queries, either to just see all the weighting factors calculated when doing the matching, or to provide a binary attribute that can be used to write a custom ranking UDF. works only with expression rankers

REMAP(condition, expression, (cond1, cond2, ...), (expr1, expr2, ...))
allows to make some exceptions of an expression values depending on condition values. Condition expression should always result integer, expression can result in integer or float.

GEODIST(lat1, lon1, lat2, lon2, { option=value, ... })

in = {deg | degrees | rad | radians}
specifies the input units

out = {m | meters | km | kilometers | ft | feet | mi | miles}
specifies the output units;

method = {adaptive | haversine}
specifies the geodistance calculation method. Default is 'adaptive', more precise and faster than 'haversine'

Cheatsheets v7 - up to Manticore Search 3.5.4
© 2017-2021 Manticore Software Ltd.
https://github.com/manticoresoftware/manticoresearch

Interested on getting help?

Check our support plans at
https://manticoresearch.com/services/

contact@manticoresearch.com

manticoresearch 3 Cheatsheets

Specific RealTime index settings

rt_mem_limit
RAM chunk size limit. Optional, default is 128M

rt_field
Full-text field. Multi-value, mandatory

rt_attr_bigint
BigInt attribute. Multi-value, optional

rt_attr_bool
Boolean attribute. Multi-value, mandatory

rt_attr_float
Float attribute. Multi-value, optional

rt_attr_json
JSON attribute. Multi-value, optional

rt_attr_multi
MVA attribute. Multi-value, optional

rt_attr_multi_64
bigInt MVA attribute. Multi-value, optional

rt_attr_string
String attribute. Multi-value, optional

rt_attr_uint
Unsigned integer attribute. Multi-value, optional

Specific Distributed index settings

agent_blackhole
Remote blackhole agent declaration in the distributed index. Multi-value, optional, default is empty.

agent_connect_timeout
Remote agent connection timeout, in milliseconds. Optional, default is 1000 (ie. 1 second).

agent_persistent
Persistently connected remote agent declaration. Multi-value, optional, default is empty.

agent_query_timeout
Remote agent query timeout, in milliseconds. Optional, default is 3000 (ie. 3 seconds).

agent_retry_count/mirror_retry_count
Integer, specifies how many times manticore will try to connect and query remote agents before report fatal query error.

agent
Remote agent declaration in the distributed index. Multi-value, optional, default is empty.

ha_strategy
Agent mirror selection strategy, for load balancing. Possible values: random(default),roundrobin,nodeads,noerrors

local
Local index declaration in the distributed index. Multi-value, optional, default is empty.

Specific Plain index settings

dict
keywords dictionary type, only for plain index, can be keywords(default) or crc(deprecated)

killlist_target
List of indexes to have kill-list applied. It can operate in 3 modes: index:k1 (kill-list is used), index:id (document ids are used), index (both kill-list and document ids are used)

infix_fields
The list of full-text fields to limit infix indexing to. Applies to dict=crc only.

inplace_enable
Whether to enable in-place index inversion. Optional, default is 0 (use separate temporary files).

inplace_hit_gap
Controls preallocated hitlist gap size. Optional, default is 0.

inplace_reloc_factor
Controls relocation buffer size within indexing memory arena. Optional, default is 0.1.

inplace_write_factor
Controls in-place write buffer size within indexing memory arena. Optional, default is 0.1.

prefix_fields
The list of full-text fields to limit prefix indexing to. Applies to dict=crc only.

source
Adds document source to local index. Multi-

Index common settings

type
Index type. Known values are plain, distributed, rt, template and percolate. Optional, default is 'plain'

path
Index files path and file name (without extension). Mandatory for plain,rt and percolate.

bigram_freq_words
List of bigram words

bigram_index
bigram index mode, can be all, first_freq or both_freq

blend_chars
Blended characters list

blend_mode
Blended tokens indexing mode, can be trim_none | trim_head | trim_tail | trim_both | skip_pure

charset_table
Accepted characters table, with case folding rules or aliases of predefined charset tables. Optional, default value is non_cjk alias.

embedded_limit
Embedded exceptions, wordforms, or stopwords file size limit. Optional, default is 16K.

exceptions
Tokenizing exceptions file. Optional, default is empty.

expand_keywords
Expand keywords with exact forms and/or stars when possible.

global_idf
The path to a file with global (cluster-wide) keyword IDFs. Optional, default is empty (use local IDFs).

hitless_words
Hitless words list. Optional, allowed values are 'all', or a list file name.

html_index_attrs
A list of markup attributes to index when stripping HTML. Optional, default is empty (do not index markup attributes).

html_remove_elements
A list of HTML elements for which to strip contents along with the elements themselves. Optional, default is empty string (do not strip contents of any elements).

html_strip
Whether to strip HTML markup from incoming full-text data. Optional, default is 0. Known values are 0 (disable stripping) and 1 (enable stripping).

ignore_chars
Ignored characters list. Optional, default is empty.

index_exact_words
Whether to index the original keywords along with the stemmed/remapped versions. Optional, default is 0 (do not index).

index_field_lengths
Enables computing and storing of field lengths (both per-document and average per-index values) into the index. Optional, default is 0 (do not compute and store).

index_sp
Whether to detect and index sentence and paragraph boundaries. Optional, default is 0 (do not detect and index).

stored_fields
A list of fields to be stored in the index.

docstore_block_size
The size of the block of documents used by document storage. Optional, default is 16kb.

docstore_compression
Compression used to compress blocks of documents in document storage.Values:'lz4', 'lz4hc' and 'none'. Optional, default is 'lz4'.

docstore_compression_level
Compression level when lz4hc is used.Values between 1-12, default 9.

Index common settings

index_zones
A list of in-field HTML/XML zones to index. Optional, default is empty (do not index zones).

max_substring_len
Maximum substring (either prefix or infix) length to index. Optional, default is 0 (do not limit indexed substrings). Applies to dict=crc only.

min_infix_len
Minimum infix prefix length to index and search. Optional, default is 0 (do not index infixes), and minimum allowed non-zero value is 2.

min_prefix_len
Minimum word prefix length to index. Optional, default is 0 (do not index prefixes).

min_stemming_len
Minimum word length at which to enable stemming. Optional, default is 1 (stem everything).

min_word_len
Minimum indexed word length. Optional, default is 1 (index everything).

morphology
A list of morphology preprocessors (stemmers or lemmatizers) to apply. Optional, default is empty (do not apply any preprocessor).

morphology_skip_fields
A list of fields there morphology preprocessors do not apply. Optional, default is empty (apply preprocessors to all fields).

ngram_chars
N-gram characters list. Optional, default is empty.

ngram_len
N-gram lengths for N-gram indexing. Optional, default is 0 (disable n-gram indexing). Known values are 0 and 1 (other lengths to be implemented).

access_blob_attrs
Control how var-length attributes (string,json,mva) are accessed, pre-read and loaded into memory. Possible values mmap, mmap_preread , mlock

access_plain_attrs
Control how numeric attributes are accessed, pre-read and loaded into memory. Possible values mmap, mmap_preread , mlock

access_doclists
Control how doclists file is accessed. Possible values file,mmap

access_hitlists
Control how hitlists file is accessed. Possible values file,mmap

overshort_step
Position increment on overshort (less that min_word_len) keywords. Optional, allowed values are 0 and 1, default is 1.

phrase_boundary
Phrase boundary characters list. Optional, default is empty.

phrase_boundary_step
Phrase boundary word position increment. Optional, default is 0.

preopen
Whether to pre-open all index files, or open them per each query. Optional, default is 0 (do not preopen).

regexp_filter
Regular expressions (regexps) to filter the fields and queries with. Optional, multi-value, default is an empty list of regexps.

stopwords
Stopword files list (space separated). Optional, default is empty.

stopword_step
Position increment on stopwords. Optional, allowed values are 0 and 1, default is 1.

stopwords_unstemmed
Whether to apply stopwords before or after stemming. Optional, default is 0 (apply stopword filter after stemming).

wordforms
Word forms dictionary. Optional, default is empty.

Cheatsheets v7 - up to Manticore Search 3.5.4
© 2017-2021 Manticore Software Ltd.
https://github.com/manticoresoftware/manticoresearch

Interested on getting help?

Check our support plans at
https://manticoresearch.com/services/

contact@manticoresearch.com

manticoresearch 3 Cheatsheets

searchd directives

agent_connect_timeout
Instance-wide defaults for agent_connect_timeout parameter.

agent_query_timeout
Instance-wide defaults for agent_query_timeout parameter

agent_retry_count
Integer, specifies how many times manticore will try to connect and query remote agents in distributed index before reporting fatal query error. Default is 0 (i.e. no retries).

agent_retry_delay
Integer, in milliseconds. Specifies the delay sphinx rest before retrying to query a remote agent in case it fails. Default is 500

attr_flush_period
Interval in seconds between flushes to disk of updated attributes data. Default is 0, flushing only occurs on daemon shutdown.

binlog_flush
Binary log transaction flush/sync mode. Optional, default is 2

0, flush and sync every second.

1, flush and sync every transaction.

2, flush every transaction, sync every second.

binlog_max_log_size
Maximum binary log file size. Optional, default is 0 (do not reopen binlog file based on size).

binlog_path
Binary log (aka transaction log) files path. Optional, default is build-time configured data directory.

client_timeout
Maximum time to wait between requests (in seconds) when using persistent connections. Optional, default is five minutes.

collation_libc_locale
Server libc locale.

collation_server
Default server collation. Optional, default is libc_ci.

data_dir
Directory path used by daemon in RT mode for storing indexes metadata,index files, replication info and binlogs

threads
Maximum number of threads the damon can use. Defaults to number of CPU cores.

docstore_cache_size
Maximum size of document blocks from document storage that are held in memory.

expansion_limit
The maximum number of expanded keywords for a single wildcard. Optional, default is 0 (no limit).

grouping_in_utc
Specifies whether timed grouping in API and SphinxQL will be calculated in local timezone, or in UTC. Optional, default is 0 (means 'local tz').

ha_period_karma
Agent mirror statistics window size, in seconds. Optional, default is 60.

ha_ping_interval
Interval between agent mirror pings, in milliseconds. Optional, default is 1000.

hostname_lookup
Hostnames renew strategy. Default is to cache at daemon start. Setting this option to 'request' disabled the caching and queries the DNS at each query.

listen_backlog
TCP listen backlog. Optional, default is 5. (Windows only)

listen
This setting lets you specify IP address and port, or Unix-domain socket path, that searchd will listen on.

```
( address ":" port | port | path ) [ ":" protocol ] [ "_vip" ]
```

listen_tfo
Allows using TCP_FASTOPEN on all listeners. Default is enabled (1)

log
Log file name. Optional, default is 'searchd.log'. All searchd run time events will be logged in this file.

max_batch_queries
Limits the amount of queries per batch. Optional, default is 32.

searchd settings (continue)

net_workers
Number of network threads for workers=thread_pool mode, default is 1.

net_wait_tm
Control busy loop interval of a network thread for workers=thread_pool mode, default is 1, might be set to -1, 0, positive integer.

net_throttle_accept net_throttle_action
Control network thread for workers=thread_pool mode, default is 0.

access_blob_attrs
Control how var-length attributes (string,json,mva) are accessed, pre-read and loaded into memory. System-wide setting. Possible values mmap, mmap_preread , mlock

access_plain_attrs
Control how numeric attributes are accessed, pre-read and loaded into memory. System-wide setting. Possible values mmap, mmap_preread , mlock

access_doclists
Control how doclists file is accessed. System-wide setting. Possible values file,mmap

access_hitlists
Control how hitlists file is accessed. System-wide setting. Possible values file,mmap

persistent_connections_limit
The maximum # of simultaneous persistent connections to remote persistent agents.

pid_file
searchd process ID file name. Mandatory.

predicted_time_costs
Costs for the query time prediction model, in nanoseconds. Optional, default is "doc=64, hit=48, skip=2048, match=64" (without the quotes).

preopen_indexes
Whether to forcibly preopen all indexes on startup. Optional, default is 1 (preopen everything).

qcache_max_bytes
Integer, in bytes. The maximum RAM allocated for cached result sets. Default is 0, meaning disabled.

qcache_thresh_msec
Integer, in milliseconds. The minimum wall time threshold for a query result to be cached. Defaults to 3000, or 3 seconds. 0 means cache everything.

qcache_ttl_sec
Integer, in seconds. The expiration period for a cached result set. Defaults to 60, or 1 minute. The minimum possible value is 1 second.

query_log_format
Query log format. Optional, allowed values are 'plain' and 'sphinxql', default is 'plain'.

query_log_min_msec
Limit (in milliseconds) that prevents the query from being written to the query log. Optional, default is 0 (all queries are written to the query log).

query_log
Query log file name. Optional, default is empty (do not log queries).

query_log_mode
Permission for log files, default is 600.

read_buffer
Per-keyword read buffer size. Optional, default is 256K.

query_log
Query log file name. Optional, default is empty (do not log queries).

query_log_mode
Permission for log files, default is 600.

jobs_queue_size
Length of the threads queue size. Default is unlimited.

read_unhinted
Unhinted read size. Optional, default is 32K.

rt_flush_period
RT indexes RAM chunk flush check period, in seconds. Optional, default is 10 hours.

rt_merge_iops
A maximum number of I/O operations (per second) that the RT chunks merge thread is allowed to start. Optional, default is 0 (no limit).

not_terms_only_allowed
Allow only negation full-text operator. Default is disable(0).

max_threads_per_query
Limit of threads open operation can use. Default is no limit.

searchd settings (continue)

rt_merge_maxiosize
A maximum size of an I/O operation that the RT chunks merge thread is allowed to start. Optional, default is 0 (no limit).

snippets_file_prefix
A prefix to prepend to the local file names when generating snippets. Optional, default is empty.

sphinxql_state
Path to a file where current SphinxQL state will be serialized.

sphinxql_timeout
Maximum time to wait between requests (in seconds) when using sphinxql interface. Optional, default is 15 minutes.

subtree_docs_cache
Max common subtree document cache size, per-query. Optional, default is 0 (disabled).

subtree_hits_cache
Max common subtree hit cache size, per-query. Optional, default is 0 (disabled).

thread_stack
Per-thread maximum stack size. Optional, default is 1M.

unlink_old
Whether to unlink .old index copies on successful rotation. Optional, default is 1 (do unlink).

watchdog
Threaded server watchdog. Optional, default is 1 (watchdog enabled).

workers
Multi-processing mode (MPM). Optional; allowed values are thread_pool, and threads. Default is thread_pool.

node_address
Specify network address of a node (used in replication clusters)

server_id
unique server identification (used in replication clusters)

mysql_version_string
A server version string to return via MySQL protocol. Optional, default is empty (return Manticore version).

seamless_rotate
Prevents searchd stalls while rotating indexes with huge amounts of data to precache. Optional, default is 1 (enable seamless rotation). On Windows systems seamless rotation is disabled by default.

shutdown_timeout
searchd –stopwait wait time, in seconds. Optional, default is 3 seconds.

max_filters
Maximum allowed per-query filter count. Optional, default is 256.

max_filter_values
Maximum allowed per-filter values count. Optional, default is 4096.

max_packet_size
Maximum allowed network packet size. Optional, default is 8M.

attr_update_reserve
Reserved space for blob attribute updates. Optional, default is 128K.

indexer settings

lemmatizer_cache
Lemmatizer cache size. Optional, default is 256K.

max_file_field_buffer
Maximum file field adaptive buffer size, bytes. Optional, default is 8 MB, minimum is 1 MB.

max_iops
Maximum I/O operations per second, for I/O throttling. Optional, default is 0 (unlimited).

max_iosize
Maximum allowed I/O operation size, in bytes, for I/O throttling. Optional, default is 0 (unlimited).

max_xmlpipe2_field
Maximum allowed field size for XMLpipe2 source type, bytes. Optional, default is 2 MB.

mem_limit
Indexing RAM usage limit. Optional, default is 128M.

on_file_field_error
How to handle IO errors in file fields. Optional, default is ignore_field.

Cheatsheets v7 - up to Manticore Search 3.5.4
© 2017-2021 Manticore Software Ltd.
https://github.com/manticoresoftware/manticoresearch

Interested on getting help?

Check our support plans at
https://manticoresearch.com/services/

contact@manticoresearch.com

manticoresearch 3 Cheatsheets

common settings

lemmatizer_base
Lemmatizer dictionaries base path.

progressive_merge
Merge Real-Time index chunks during OPTIMIZE operation from smaller to bigger. Enabled by default. If disabled, chunks are merged from first to last created.

json_autoconv_keynames
Whether and how to auto-convert key names within JSON attributes. Known value is 'lowercase'. Optional, default value is unspecified (do not convert anything).

json_autoconv_numbers
Automatically detect and convert possible JSON strings that represent numbers, into numeric attributes. Optional, default value is 0 (do not convert strings into numbers).

on_json_attr_error
What to do if JSON format errors are found. Optional, default value is ignore_attr (ignore errors). Applies only to sql_attr_json attributes.

plugin_dir
Trusted location for the dynamic libraries (UDFs). Optional, default is empty (no location).

HTTP API

/insert
creates new document

/replace
replace existing document

New updates are put first in a fixed size memory called RAM chunk

/delete
delete existing documents

/update
update attributes of a document

/bulk
Allows bulking /insert,/replace,delete. Request body must be in NDJSON format

/search
Perform searches

/pq
Perform percolate queries

/sql
Proxy SQL queries

Data types (for CREATE/ALTER)

text = text indexed stored
full-text field (rt_field + stored_fields)

text indexed
non-stored full-text field (rt_field)

text stored
stored, but not indexed field (rt_field + stored_only_fields)

integer
integer attribute (rt_attr_uint)

float
float attribute (rt_attr_float)

multi
MVA attribute (rt_attr_multi)

multi64
64bit MVA attribute (rt_attr_multi64)

bool
bool attribute (rt_attr_bool)

json
json attribute (rt_attr_json)

string
string attribute (rt_attr_string). Accept indexed,attribute options.

timestamp
timestamp attribute (rt_attr_timestamp)

bit(n)
Bitcount integer attribute (rt_attr_uint)

Common Searchd daemon usage

It is recommended to install and run the searchd as a service to ensure in case of reboots, searchd will start at boot.

searchd should run under a specific user. Linux packages create specific user (manticore or manticoresearch), as well as they install a service file

```
$ service manticore start
$ service manticore stop
```

or

```
$ systemctl start manticore
$ systemctl stop manticore
```

Searchd can also be run manually.Here are several common commands used:

`$ searchd`
start searchd using a defined configuration file. It start daemonized, except Windows.

`$ searchd --stopwait`
Synced stop

--config /path/to/sphinx.conf
Specify config file. In absence of it, depending on how was compiled, searchd can try to locate the config file in current folder or a specific folder set at compilation

`$ searchd --status`
Prints status and performance counters (same as SHOW STATUS)

`$ searchd --logdebug|--logdebugv|logdebugvv`
Enabled additional debug output in daemon log

`$ searchd --iostats --cpustats`
Provide io/cpu counter stats

`$ searchd --install`
Install searchd as service on Windows

Searchd supports receiving signals.

`$ kill -TERM cat /var/run/searchd.pid`
Sends a shutdown signal

`$ kill -HUP cat /var/run/searchd.pid`
Initiate index rotation. This will also reload the configuration and can be used to activate new indexes (like a new distributed index).

`$ kill -USR1 cat /var/run/searchd.pid`
Force reopen of log files, useful for implemention log file rotation

Quick facts about index types

plain
immutable text data.Require full reindexing. Attributes can be updated.

Can use one or more defined sources

Can be converted to a RealTime index

RealTime
Starts empty,data can be added/changed/deleted similar to a SQL table.New updates are put first in a fixed size memory called RAM chunk

Supports autogenerated IDs

When filled, the RAM chunk is discarded to disk as a disk chunk (which is almost identical as a plain index)

As disk chunk increase, performance is affected and OPTIMIZE can be used to reduce no. of chunk

Search can be multithreaded for RT with multiple chunks

distributed
Holds no data

Acts as master to local and/or remote indexes, by sending queries to them and merging back the results.

In case of locals, threads needs to be used to allow creating parallel query threads

supports mirrors of remote indexes and load balancing

template
Holds no data

Can be used to test tokenization rules or highlighting incoming data

percolate
Stores queries and not documents.It's based on RealTime type.

Common Indexer daemon usage

Indexer should run under the same user the searchd runs. This is to make sure searchd can rotate indexes created by indexer.

`$ indexer --config /path/to/sphinx.conf index --rotate`
The most common use of indexer is to issue an indexing followed by a rotation, while searchd is running

Is searchd is not running, parameter --rotate should be omitted. The new index version will replace existing one at searchd startup.

Indexer can be used to extract a dictionary of most common words from an index, using --buildstops. --buildfreqs provides additionally the quantity presence.

```
$ indexer myindex --buildstops word_freq.txt 1000 --buildfreqs
```

Indexer can merge two indexes into one. This can be used in case of main+delta setups

```
$ indexer --merge main delta --rotate
```

Other useful parameters include --print-queries for printing the SQL queries run by a plain index with SQL sourcem --dump-rows to dump the fetched rows from a SQL source to a file and --print-rt for printings inserts ready for a RT index

Configuration file

Config use a custom plain format. Each section starts with name and the list of directives are enclosed by curly brackets.

Sections:

searchd
Holds searchd settings. Only one section allowed

indexer
Holds indexer settings. Only one section allowed

common
Holds several settings common to searchd and indexer. Only one section allowed

index
Index definition section.Multiple allowed

source
Plain index source definiton. Multiple allowed

Operating modes

Starting with 3.4 the daemon can operate in 2 modes:

RT mode
Configuration holds no indexes. Indexes can be created/dropped with statements. Plain and template indexes not supported. Replication supported only in this mode. Default since 3.4.

Plain mode
Indexes defined in configuration file. Replication not supported. Legacy mode, recommended if plain indexes are required or migration to RT mode is difficult.

Connecting using SphinxQL interface

`$ mysql -P9306 -h0`
or

`$ mysql -P9306 -h127.0.0.1`
retrieve the warning produced by the latest query

No credentials are needed. Database selection is not needed, `USE dbname;` is a dummy command in Manticore.

SphinxQL use the MySQL as transport protocol, but doesn't implement the full set of MySQL commands.

Cheatsheets v7 - up to Manticore Search 3.5.4
© 2017-2021 Manticore Software Ltd.
https://github.com/manticoresoftware/manticoresearch

Interested on getting help?

Check our support plans at
https://manticoresearch.com/services/

contact@manticoresearch.com