RediSearch 1.6 release notes

Requirements

RediSearch v1.6.16 requires:

- Minimum Redis compatibility version (database): 4.0.0
- Minimum Redis Enterprise Software version (cluster): 5.0.0

v1.6.16 (June 2021)

This is a maintenance release for version 1.6.

Update urgency: MODERATE - Program an upgrade of the server, but it’s not urgent.

Details:

- Bug fix:
  - #2018: FT.ADD REPLACE leaves fields from the previous document that are not included in the new document
  #647 #1193

v1.6.15 (February 2021)

This is a maintenance release for version 1.6.

Update urgency: Low

Details:

- Minor enhancements:
  - #1225 Allow scientific representation of numbers for numeric fields.
  - #1574 Allow SORTBY for non-sortable fields.

- Minor bugfixes:
  - #1683 Add a module parameter _NUMERIC_COMPRESS which prevent double -> float compression. It prevents an issue where an exact match on some floating-point numbers is not found.
  - #1757 Remove assertion on the limitation of the number of sortable fields, instead return an error.
  - #1668 Query words from stopword list on tag fields.
  - #1745 Enforce 0 value for count reducer.
  - #1774 MINPREFIX & MAXEXPANSION can be changed in runtime.
  - #1861 Fix issue for FT.SCOREEXPLAIN where reply array depth can exceed 7.
  - #1689 FT.SUGGET results from RSCoordinator are more consistent.
  - Various small tweaks under the hood.

v1.6.14 (September 2020)

- This is a maintenance release for version 1.6.

Details:
• Minor features:
  • #1420 The hard limit of the number of results produced by FT.SEARCH is now configurable with MAXSEARCHRESULTS.

• Bug fixes:
  • #1313 Wrong error of unsupported phonetic field.
  • #1286 Possible crash on optional search.
  • #1449 Rare file descriptor leak on FORK GC.
  • #1469 Endless loop when reaching internal docid above uint32_max.

v1.6.13 (May 2020)
This is a maintenance release for version 1.6.

Headlines:
• This release improves overall stability and provides fixes for issues found after the previous release.

Details:
• Bugfixes:
  • #1186 #1188 incorrect values for inverted_sz_mb and num_records in FT.INFO command.

v1.6.12 (April 2020)

Headlines:
• This release improves overall stability and provides fixes for issues found after the previous release.

Details:
• Features:
  • #1172 Added exists function that can be used on conditional updates (REPLACE PARTIAL) to check if a field exists in the document.

• Minor Enhancements:
  • #1172 Lazy evaluation of the right side of ‘or’/‘and’ clauses in IF condition.
  • #1134 Remove hard limit on LIMIT when using FT.SEARCH.

• Bugfixes:
  • #1124 NOINDEX tag fields could not be updated on UPDATE PARTIAL with no indexed fields.
  • #1120 Release loop in II_GetCriteriaTester which released the same criteria tester multiple times.
  • #1161 Case where setting MAXDOCTABLESIZE had no effect.
  • #1169 FIRST_VALUE reducer crashed when value did not exist.
  • #1159 Infinite loop on NOT criteria tester.

v1.6.11 (March 2020)

Headlines:
• This release improves overall stability and provides fixes for issues found after the previous release.

Details:
• Bugfixes:
• #1126 Memory leak introduced by queries for tag fields that have no results.

v1.6.10 (March 2020)

Headlines:

• This release improves overall stability and provides fixes for issues found after the previous release.
• This release fixes certain backwards compatibility issues compared to 1.4. Although they are rare cases, it is recommended when upgrading to 1.6 to use this version or newer.

Details:

• Minor Enhancements:
  • #1062 Added Custom stopwords list in FT.INFO

• Fixed backwards incompatible issues:
  • #1075 Fields should always be returned to the user as a string.
  • #1074 Don’t truncate possible integral values when printing.
  • #1065 Revert “Change how generated reducer aliases are made”.

• Bugfixes:
  • #1085 Min and max value on non leaf nodes in the numeric tree should not be updated.
  • #1106 Pipe leak on FORK GC caused by closing the fork without holding the lock.
  • #1114 PR #986 reverted the work from #985, #989. This PR reintroduces these features.

v1.6.9 (February 2020)

Headlines:

• This release improves overall stability and provides fixes for issues found after the previous release.

Details:

• Bugfixes:
  • #1052 Remove wrong optimization on Quantile.
  • #1057 Memory pool did not release memory when certain limit was reached.

v1.6.8 (February 2020)

Headlines:

• This release improves overall stability and provides fixes for issues found after the previous release.

Details:

• Bugfixes:
  • #1052 Remove wrong optimization on Quantile.
  • #1057 Memory pool did not release memory when certain limit was reached.

v1.6 GA (January 2020)

This is the General Availability Release of RediSearch 1.6 (v1.6.7).

Headlines:
Several performance improvements increasing full-text search queries up to 60% and aggregation queries up to 73%.

Support for aliasing of indices.

Low-level API in C (and Rust bindings) to make RediSearch embeddable in other Redis modules. RedisGraph is the first GA consumer.

Forked process Garbage Collection (FORK GC) allows for stable read latencies.

Full details:

**Added functionality**

- **#658** FT.ADD REPLACE NOCREATE will not add the document if the document does not exist.
- **#575** Add index aliasing. This allows users to provide (or remove) 'links' to indexes. The commands are FT.ALIASADD, FT.ALIASDEL, and FT.ALIASUPDATE. See documentation for full details
- New C API to make RediSearch embeddable in other Redis modules. This API allows other Redis modules to use functionality of RediSearch without actually having the "module" functionality active. Note that this must still be used on Redis proper. Modules that are already incorporating this API RedisGraph GA
  - RedisTimeSeries (WIP)
  - RedisJSON (WIP)

**Performance improvements**

- Improve performance when using many union (|) iterators
- Improve performance when using many intersect iterators
- Improve overall index reading performance
- **#598** Do not return foo: NULL if foo is not present in the document. This conserves network bandwidth

**Bugfixes - Semantics**

- **#688 #623** Fix various issues with optional (~) search operator:
  - Fixes omitted results when using union operators in addition to optional iterators.
  - Allow optional iterator to be used in isolation in promotion-only mode (without a filter query)
  - Fix issue where weight attribute was being ignored
- **#653** FT.GET will no longer return a document as existing if it was not added by FT.ADD, even if the document exists in the server as a plain redis hash
  - FT.AGGREGATE is now more stringent with its semantics, avoiding nonsensical queries or referencing fields which do not exist in the schema or LOADED.
- **#779** Added to_number() and to_str() functions for ambiguity reasons
- **#906** A description of how scores were calculated can be added by adding 'EXPLAINSCORE'
- **#897** FORK GC has now lowest priority over indexing and read queries
  - Added automated tests to ensure macOS build works

**Bugfixes - Crash/Stability**

- **#666** Fix crash when conflict between internal key name and user key name is encountered; e.g. creating a new document with ft:two/two
- **#697 #588** Fix memory leaks
- **#691** Fix crash on FT.EXPLAIN
  - Proper module-level and index-level cleanup functionality
  - Simplified concurrency model
- **#898** Fix rare issue where FORK GC doesn't exist on termination of Redis
- **#865** When using FT.SEARCH with SORTBY, it will only be possible to sort by one field
- **#917** Fix wrong results introduced in a skip optimisation
- **#888** NULL terms cause FORK GC to crash
- **#887** Chinese searches not being converted to simplified Chinese.
- Fix FORK GC issue where read from pipe did not returned all the data