6.1.2 Multiple keywords

Note

This page contains some advice on generating multiple keyword index tables. However, please note the wide availability of open source search tools and search capabilities built into databases makes this advice less relevant than it was when originally issued.

The performance of single keyword searches is highly dependent on the number of candidate descriptions returned by the keyword for subsequent filtering. The extremely high number of matches for some words in common use makes it likely that some searches will be unacceptably slow.

One way to alleviate this problem is to create a table containing a row for all combinations of word pairs in each description. In some database environments that support optimization of multiple key searches, this may offer no benefits. However, in other environments, such a table may substantially speed searches.

A comprehensive word pair table would be very large. Such a table covering the full content of SNOMED CT would contain approximately 1.5 million unique word pairs and 6 million rows. Limiting the unique keys to the first three letter of each word reduces the table size to a more readily optimized set of keys. This requires the final part of the search to be conducted using text comparison (since the keys are incomplete).

Generating a dual key index

For each description, parse the text of the term:

- To avoid inappropriate case mismatches, convert all characters to the same case;
- Extract words by breaking at spaces, punctuation marks, and brackets;
- For each word of three characters or more that is not in a list of excluded words, extract the first 3 characters, and arrange the word fragments in alphabetical order;
- Generate the dual keys for this description by concatenating each word fragment with those that come after it in the list;
- For each dual key, add a row to the word pair tables.

Example: Generation of dual keywords for a sample description

Table 6.1.2-1: Example Description

Description Identifier	Concept Identifier	Term
	19954002 |Reconstruction of hip with use of methyl methacrylate (procedure)|	Total replacement of hip with use of methyl methacrylate

To avoid inappropriate case mismatches, convert all characters to the same case

"TOTAL REPLACEMENT OF HIP WITH USE OF METHYL METHACRYLATE"

Extract words by breaking at spaces, punctuation marks, and brackets

1. TOTAL;
2. REPLACEMENT;
3. OF;
4. HIP;
5. WITH;
6. USE;
7. OF;
8. METHYL;
9. METHACRYLATE.

For each word of three characters or more, that is not in a list of excluded words, extract the first 3 characters, and arrange the word fragments in alphabetical order.

1. HIP;
2. MET;
3. REP;
4. TOT;
5. USE.

⚠ In this example "OF" and "WITH" are excluded as they are in a list of excluded words, while "MET" is duplicated, so we only include it once.

Generate the dual keys for this description by concatenating each word fragment with those that come after it in the list

For each dual key, add rows to the word pair tables

Table 6.1.2-2: Example Dual Key Index

Dual key	Description Identifier
HIPMET	33592011
HIPREP	33592011
HIPTOT	33592011
HIPUSE	33592011
METREP	33592011
METTOT	33592011
METUSE	33592011
REPTOT	33592011
REPUSE	33592011
TOTUSE	33592011

Searching for descriptions using a dual key index

A search on the dual key index can only be carried out if the user enters a search string that contains at least two word fragments both of which are three characters or more in length. If the search string does not meet this criterion, the single keyword search mechanism must be used.

- The user-typed search string is converted to consistent case;
- The string is parsed, breaking at spaces and punctuation characters;
- For each word of three characters or more, extract the first 3 characters, and arrange the word fragments in alphabetical order ;
- Create a dual key by concatenating the first two 3 letter word fragments;
- Use this dual key to look up exact matches on the word pair index;
- Descriptions found by searching on the word pair index are screened, to see if they contain the complete words in the original search string.

Example: Search using word pair index

User searches for "PYRO* 1 OXYGEN*".

The string is parsed, breaking at spaces and punctuation characters.

1. "PYRO*";
2. 1;
3. "OXYGEN*".

For each word of three characters or more, extract the first 3 characters, and arrange the word fragments in alphabetical order .

1. "OXY";
2. "PYR".

Create a dual key by concatenating the first two 3 letter word fragments.

- OXYPYR

Use this dual key to look up exact matches on the word pair index.

Table 6.1.2-3: Sample results of a search for "PYRO* 1 OXYGEN*"

Dual key	Description Identifier	Description
OXYPYR	1969019	104951019 |2,5-Dihydroxy-pyridine oxygenase|
OXYPYR	22565018	22565018 |pyrogallol 1,2-oxygenase|
OXYPRY	104951019	104951019 |2,5-Dihydroxy-pyridine oxygenase|

- Descriptions found by searching on the word pair index are screened, to see if they contain the complete words in the original search string:
  - Description 1969019 is eliminated since it does not contain the word "1";
  - Description 104951019 is eliminated, it does not contain the word "1" or any word beginning with the string "pyro".