Introduction to MySQL natural language search

In natural language searches, MySQL looks for rows or documents that are relevant to the free-text natural human language query, for example, “How to use MySQL full-text search”.

Relevance is a positive floating-point number. When the relevance is zero, it means that there is no similarity. MySQL computes the relevance based on various factors including the number of words in the record, the number of unique words in the record, the total number of words in the collection, and the number of documents (rows) that contain a particular word.

To perform natural language searches, you use MATCH()  and  AGAINST() functions. The MATCH() function specifies the column where you want to search and the AGAINST()  function determines the search expression to be used.

MySQL natural language search example

We will use the products table in the [sample database](http://www.mysqltutorial.org/mysql-sample-database.aspx) for the demonstration.

First, we need to enable full-text search in the productLine  column of the products  table using theALTER TABLE ADD FULLTEXT  statement:

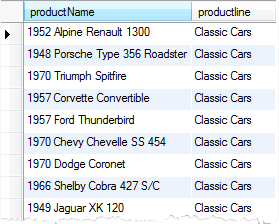
MySQL

|  |  |
| --- | --- |
| 1  2 | ALTER TABLE products  ADD FULLTEXT(productline) |

Second, we can search for products whose product lines have a word Classic . We use the MATCH() and AGAINST()  functions as the following query:

MySQL

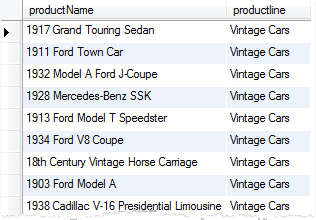
|  |  |
| --- | --- |
| 1  2  3 | SELECT productName, productline  FROM products  WHERE MATCH(productline) AGAINST('Classic') |



To search for product whose product line contains Classic or Vintage  keyword, you can perform the following query:

MySQL

|  |  |
| --- | --- |
| 1  2  3 | SELECT productName, productline  FROM products  WHERE MATCH(productline) AGAINST('Classic,Vintage') |

  
The AGAINST()  function uses IN NATURAL LANGUAGE MODE  search modifier by default therefore you can omit it in the query. There are other search modifiers e.g.,  IN BOOLEAN MODE   for [Boolean text searches](http://www.mysqltutorial.org/mysql-boolean-text-searches.aspx).

You can explicitly use the IN NATURAL LANGUAGE MODE  search modifier in your query as follows:

MySQL

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT productName, productline  FROM products  WHERE MATCH(productline)  AGAINST('Classic,Vintage' IN NATURAL LANGUAGE MODE) |

By default, MySQL performs searches in the case-insensitive fashion. However, you can instruct MySQL to perform case-sensitive searches using binary collation for indexed columns.

Sort the result set by relevance

A very important feature of full-text search is how MySQL ranks the rows in the result set based on their relevance. When the MATCH()  function is used in the [WHERE clause](http://www.mysqltutorial.org/mysql-where/), MySQL returns the rows that are more relevant first.

The following example shows you how MySQL sorts the result set by the relevance.

First, we enable the full-text search feature for the  productName column of the products table.

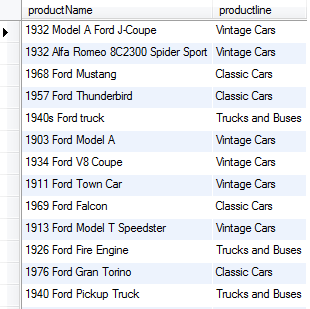
MySQL

|  |  |
| --- | --- |
| 1  2 | ALTER TABLE products  ADD FULLTEXT(productName) |

Second, we search for products whose names contain  Ford   and/or  1932 using the following query:

MySQL

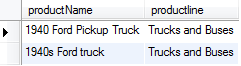
|  |  |
| --- | --- |
| 1  2  3 | SELECT productName, productline  FROM products  WHERE MATCH(productName) AGAINST('1932,Ford') |

Let’s examine the output:  
  
The products, whose names contain both 1932  and Ford are returned first and then the products whose names contains the only Ford  keyword.

To perform a full-text search in the Boolean mode, you use the IN BOOLEAN MODE modifier in theAGAINST  expression. The following example shows you how to search for a product whose product name contains the Truck word.

MySQL

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT productName, productline  FROM products  WHERE MATCH(productName)        AGAINST('Truck' IN BOOLEAN MODE ) |

  
Two products whose product names contain the Truck  word are returned.

To find the product whose product names contain the   Truck word but not any rows that contain  Pickup , you can use the exclude Boolean operator ( - ), which returns the result that excludes the Pickup keyword as the following query:

MySQL

|  |  |
| --- | --- |
| 1  2  3 | SELECT productName, productline  FROM products  WHERE MATCH(productName) AGAINST('Truck -Pickup' IN BOOLEAN MODE ) |

mysql boolean tex searches with Boolean operator