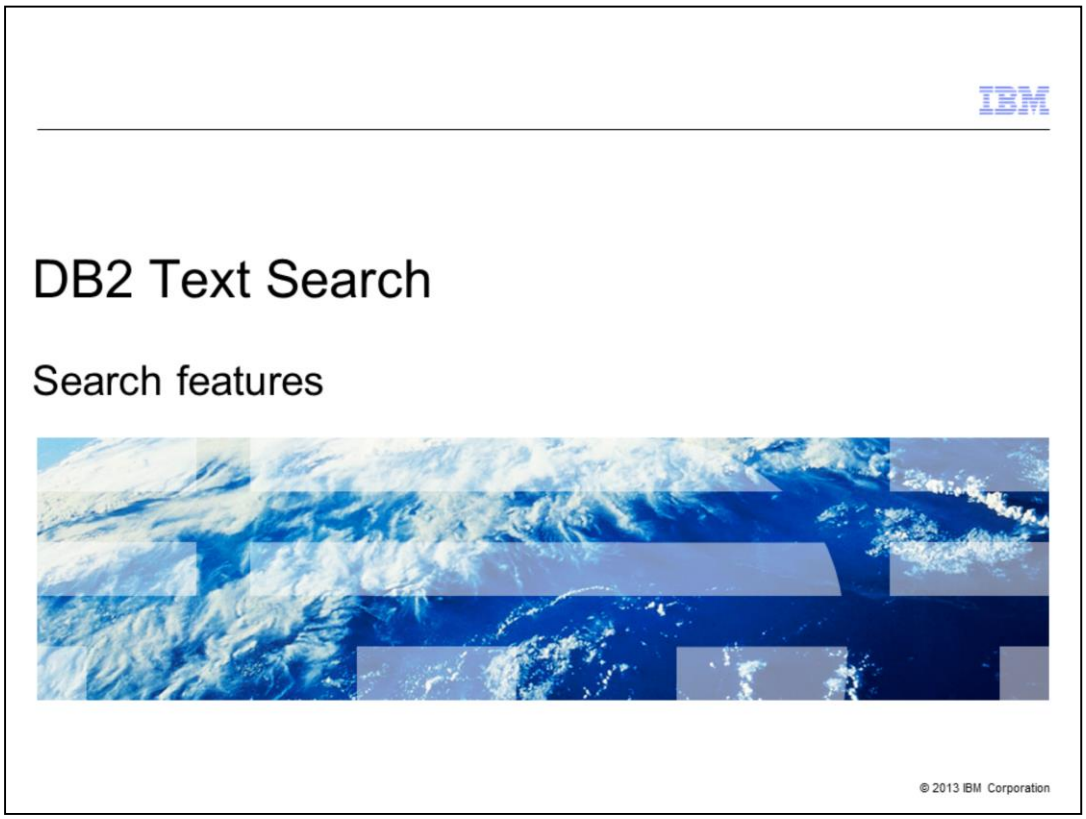
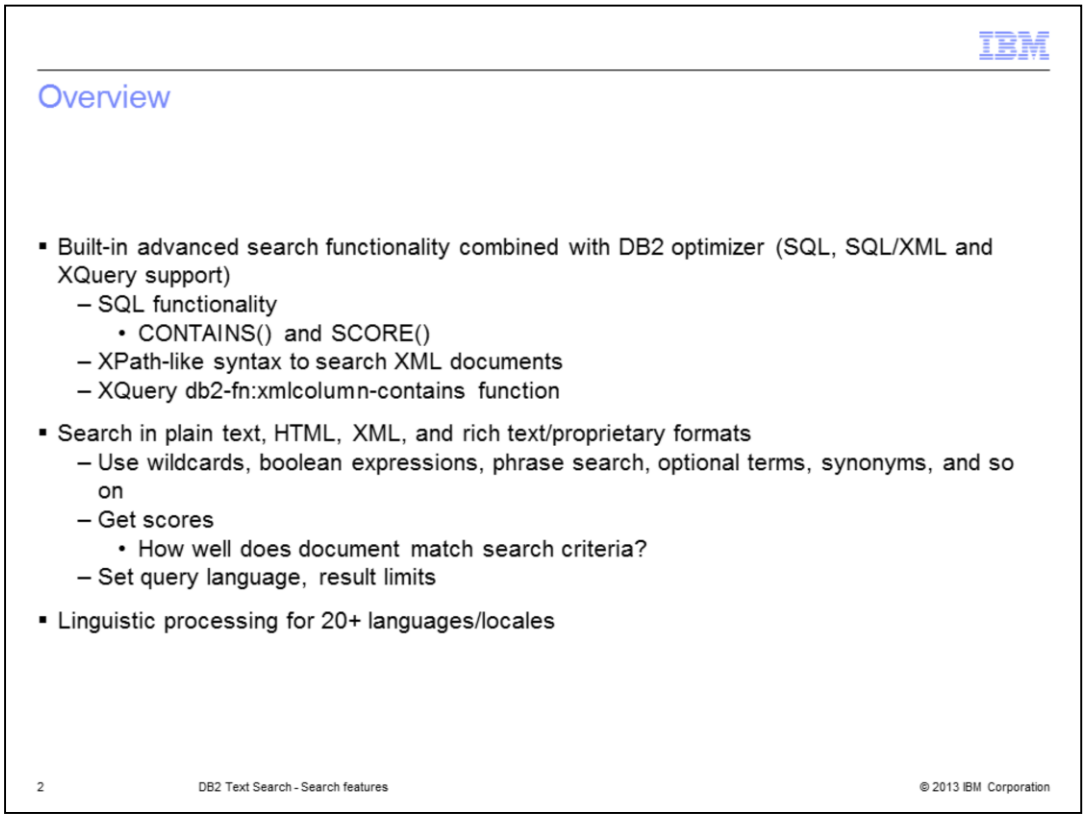


This presentation describes various search features available in DB2® Text Search.



DB2 Text Search allows you to issue SQL and XQuery statements to search the data stored in your DB2 database. The built-in SQL functions, CONTAINS() and SCORE(), are used to search the text indexes and are seamlessly integrated with DB2 optimizer to automatically select the best optimization plan according to the expected search results.

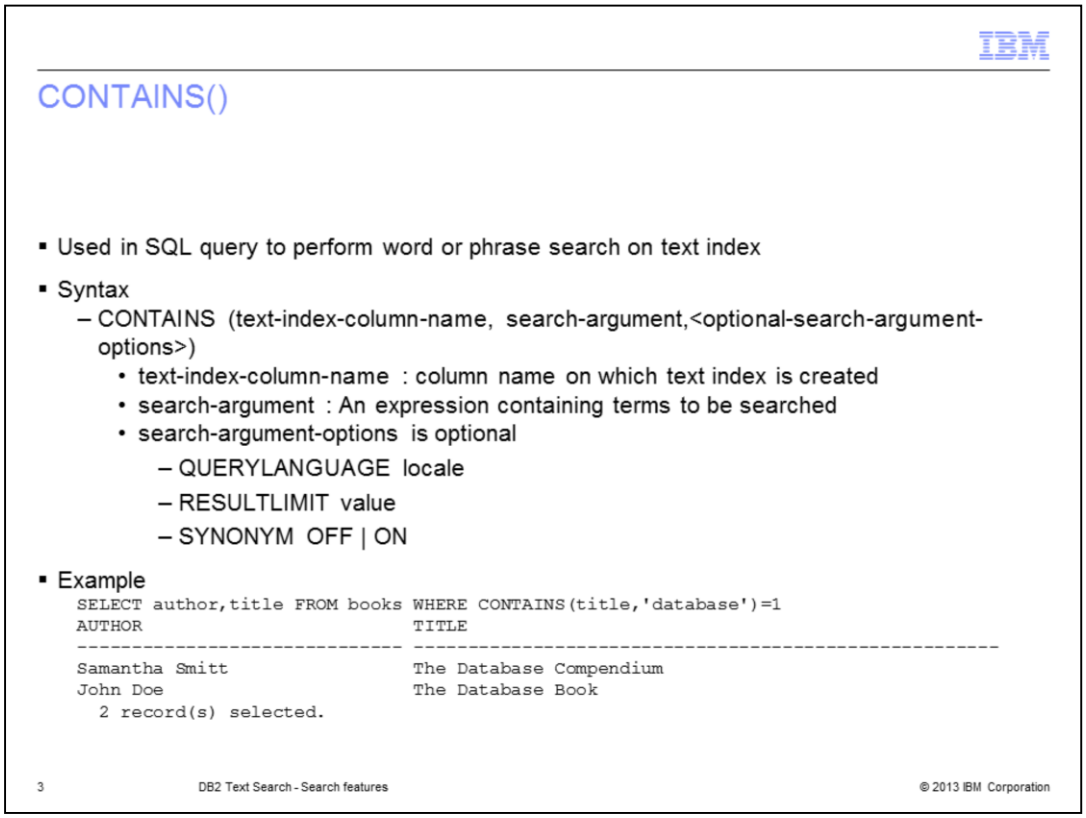
Using the XPath syntax subset and the built in search function db2-fn:xmlcolumn-contains(),it is possible to perform a structural full text search in XML documents. It allows you to search specific portions of an XML document.

Supported document formats include Text, HTML, XML and, with the DB2 Accessories Suite, rich text documents are searchable.

The search scope can either be extended or reduced using wild cards, Boolean expressions, phrase search, optional terms, synonyms and so on. Using the SCORE() function, you can find out how well a document meets the search criteria compared to other documents in the same text index. Other search features include setting the query language in the search query to explicitly specify the locale to DB2 Text Search, result limit can be specified to limit the returned result set.

DB2 Text search also supports linguistic processing for 20+ languages to retrieve only the relevant search results.

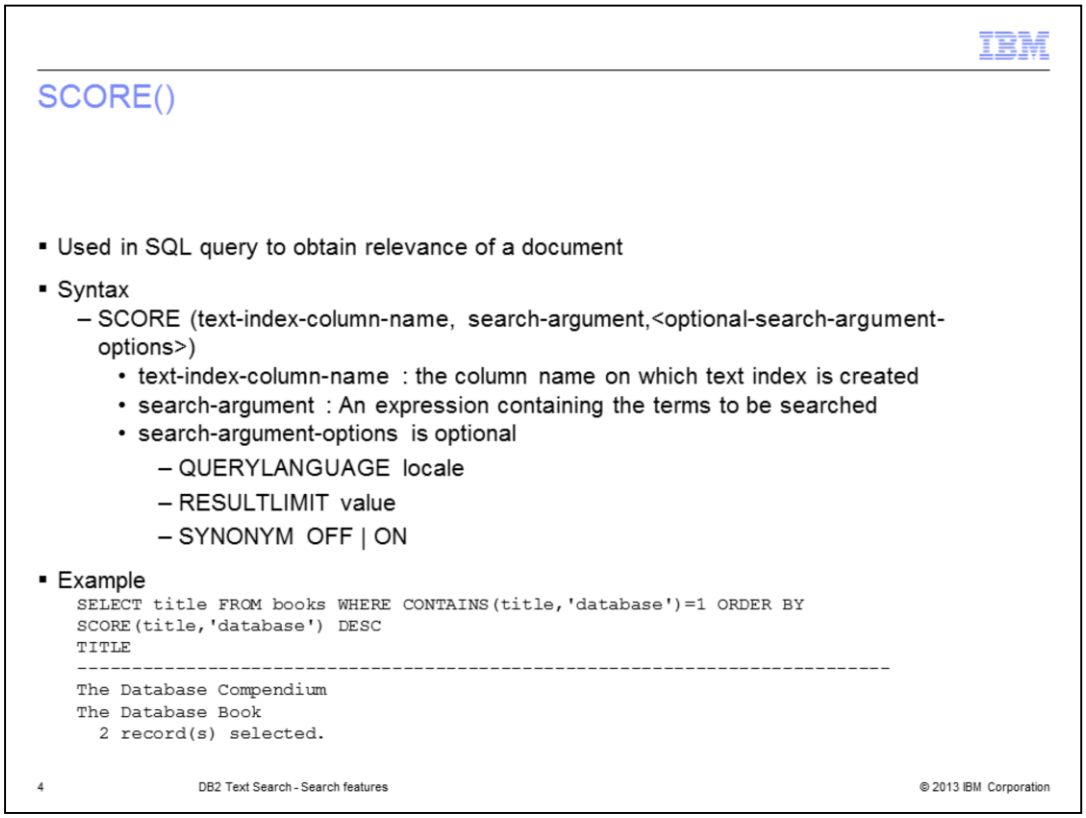
The next slides provide more details and examples about the search functionality.



The SQL function CONTAINS() searches the text search index using the criteria specified in the search query. It returns an integer value of one if the document contains the text, otherwise it returns zero. This is the most common way of performing a search with DB2 Text Search. Use this where standard SQL is used and combine it with other conditions using the WHERE clause.

The CONTAINS() function takes two mandatory parameters; the column name on which the text index is created and the actual search criteria that should be applied. There is a third optional parameter used to specify the options for search criteria.

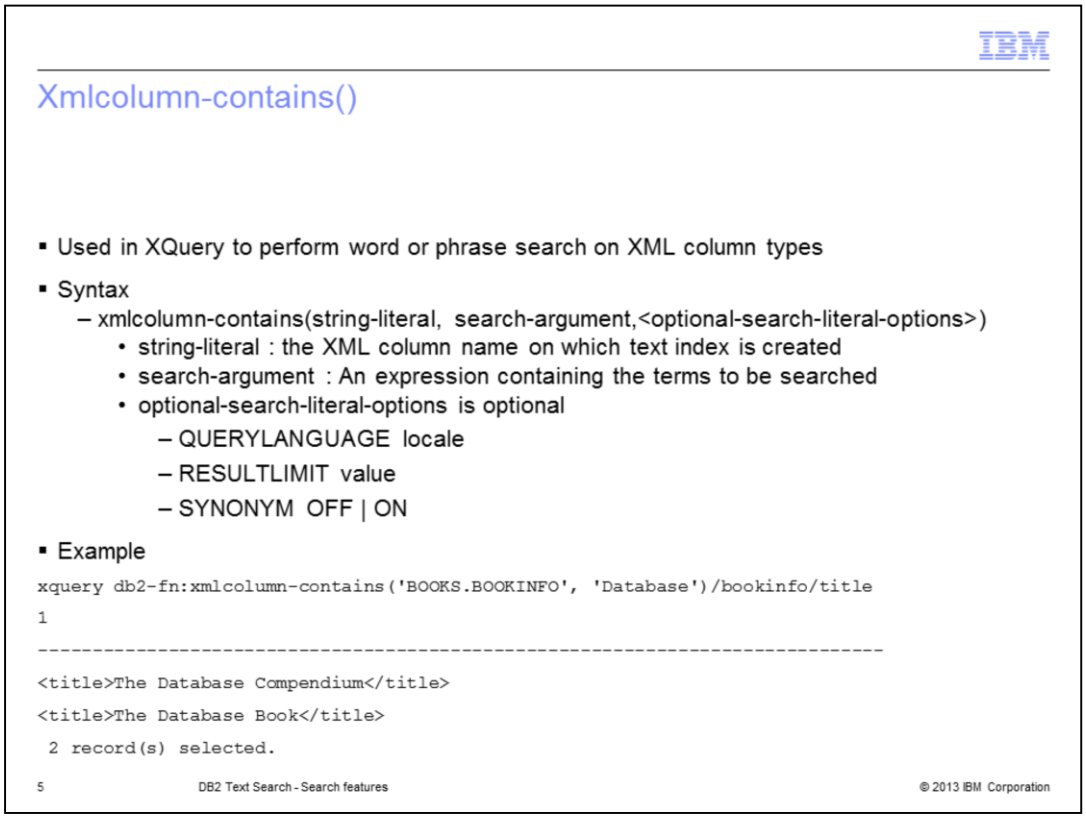
There is an example provided on this slide to show how to search the ‘title’ column in the ‘books’ table for a specific search term ‘database’ using the CONTAINS() SQL function.



The SQL function SCORE() can be used to obtain the relevancy of a found text document. It returns a double precision floating-point number between zero and one.

The SCORE() function takes two mandatory parameters; the column name on which the text index is created and the actual search criteria that should be applied. There is a third optional parameter used to specify the options for search criteria.

There is an example displayed on this slide to show how to use the SCORE() function to find out which documents qualifies the search criteria better compared to the others.



The XML function xmlcolumn-contains() can be used in XQuery statement and operates on XML columns only. It returns XML documents containing the search term or phrase.

This function takes two mandatory parameters; the column name on which the text index is created and the actual search criteria that should be applied. There is a third optional parameter used to specify the options for search criteria.

There is an example displayed on this slide to show how to use the db2-fn:xmlcolumn-contains function in an XQuery to find the XML documents stored in the column ‘books.bookinfo’ that contain the search term ‘Database’.

Search XML documents

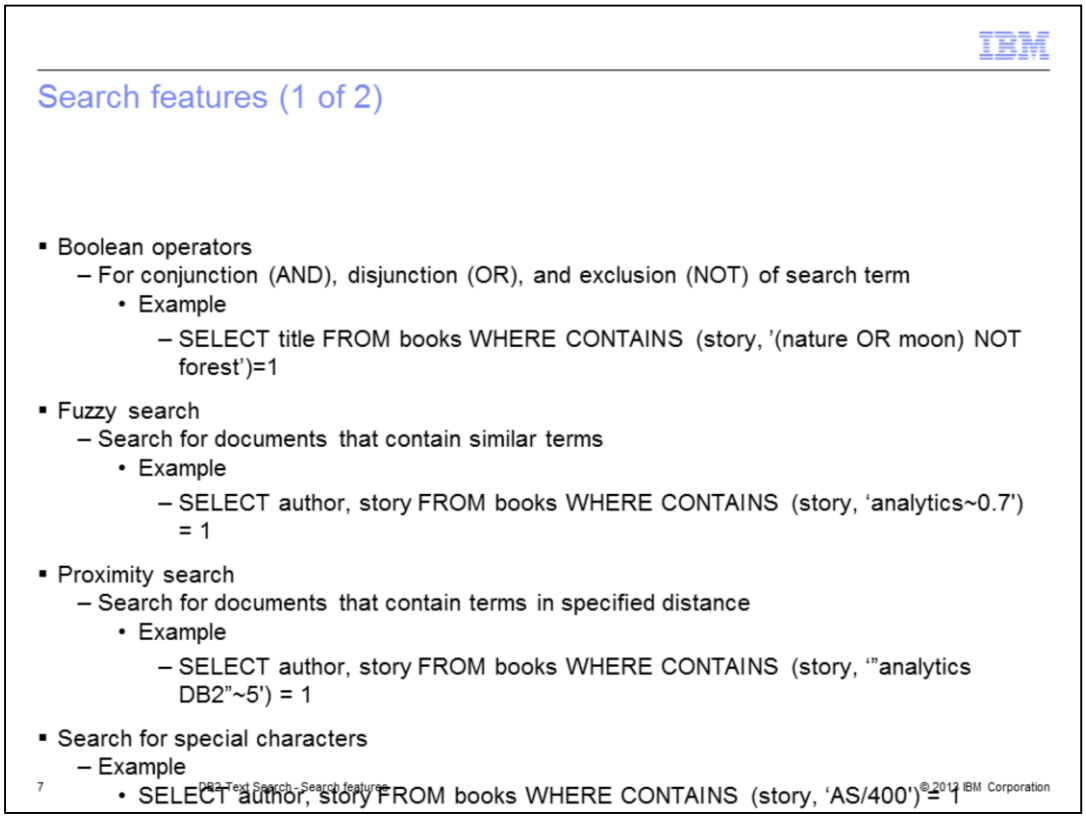
- Example

```
SELECT author FROM books WHERE
contains(bookinfo, '@xpath:''/bookinfo/story [.contains("Database")]''')=1

AUTHOR
------------------------------
Samantha Smitt
John Doe

  2 record(s) selected.
```

- Example

```
xquery db2-fn:xmlcolumn-contains
( 'BOOKS.BOOKINFO','@xpath:''/bookinfo/story[.contains("Database")]''')/bookinfo/title
1
------------------------------------------------------------
<title>The Database Compendium</title>
<title>The Database Book</title>
 2 record(s) selected.
```


The examples displayed on this slide show two different ways of searching XML documents using the XPath expression in CONTAINS() SQL function and db2-fn:xmlcolumn-contains() XML function in XQuery context.



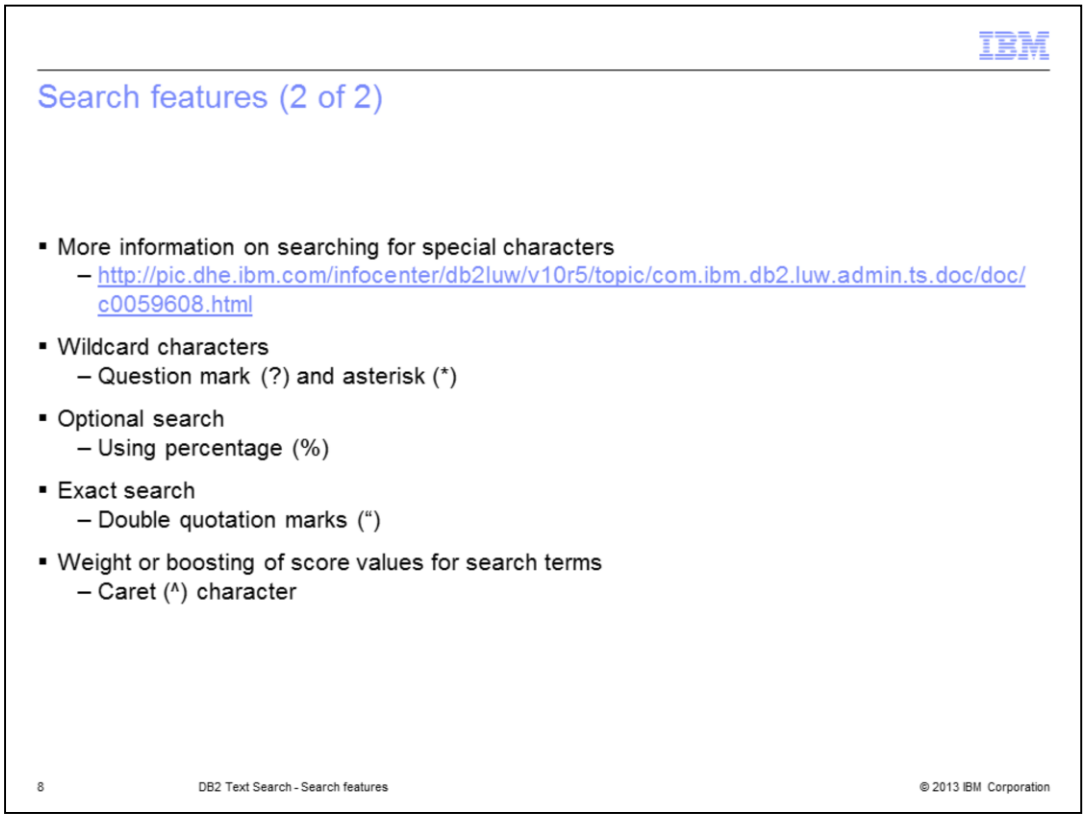
Now that you have some understanding about the basic search functionality, the next few slides discusses other search features offered by DB2 Text Search.

You can use the boolean operators – AND and OR to combine several search terms. You can use the boolean operator "NOT" to exclude any particular text document from the search. The example displayed on this slide searches for documents that contain the terms ‘nature’ or ‘moon’ but not ‘forest.

You can use the Fuzzy search to find the documents that contain words which are spelled in a similar way to the search term. Use the tilde symbol at the end of a term to do a Fuzzy search. The example displayed on this slide finds the documents that include the terms analytics, analysis, analyze, and so on. The fuzzy factor is 0.7 in this example, increase it further to include more results.

You can use the Proximity search to find the documents that contain terms in a distance of at most, x words. Use the tilde (~) symbol at the end of a phrase and specify the distance in words as a valid integer number. The example displayed on this slide finds the documents with terms ‘analytics’ and ‘DB2’ located within five words from each other.

The special characters can be searched in the same way as the other query terms. Some of them should be escaped in order to be searched.



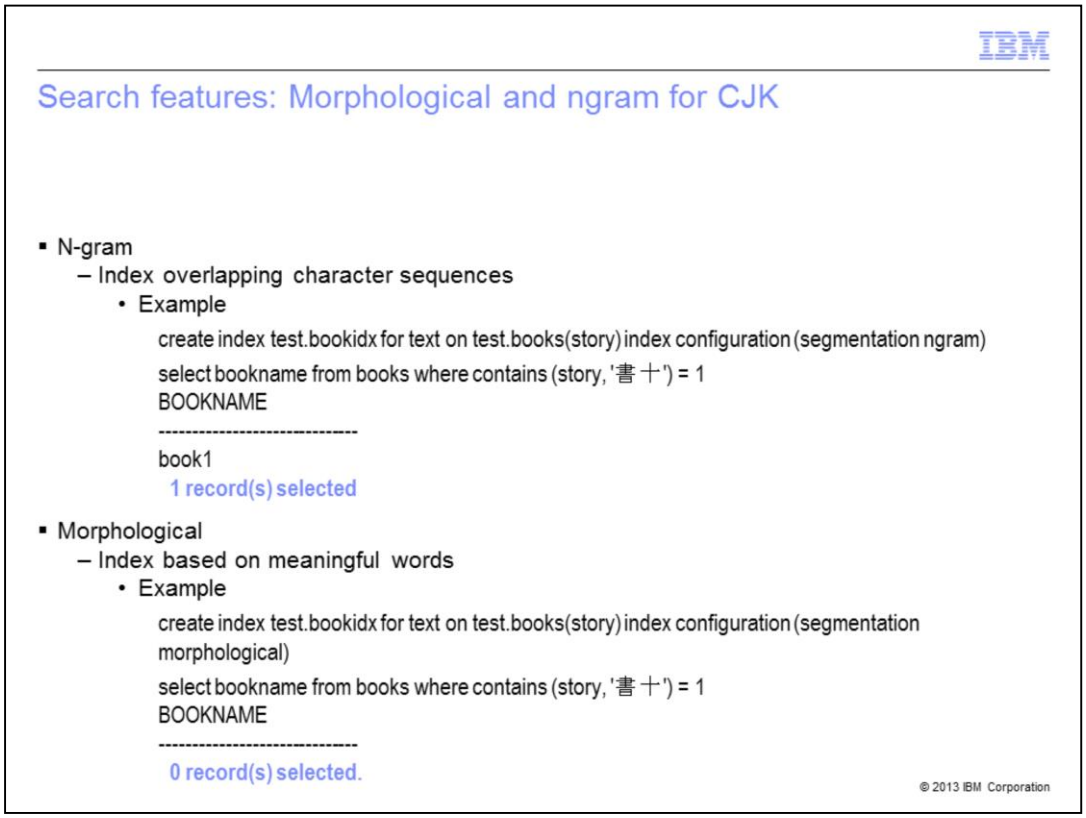
For more information on searching for special characters, see the link displayed on this slide.

Increase the search scope by using wildcard characters. Use a question mark (?) to specify that a single character can be added to your search term. Use an asterisk (*) to specify that any number of characters can be added to your search term.

You can use a percentage sign (%) to specify that a term or phrase is optional while executing the search.

For an exact search, use quotation marks (") around your search term or phrase to perform an exact search.

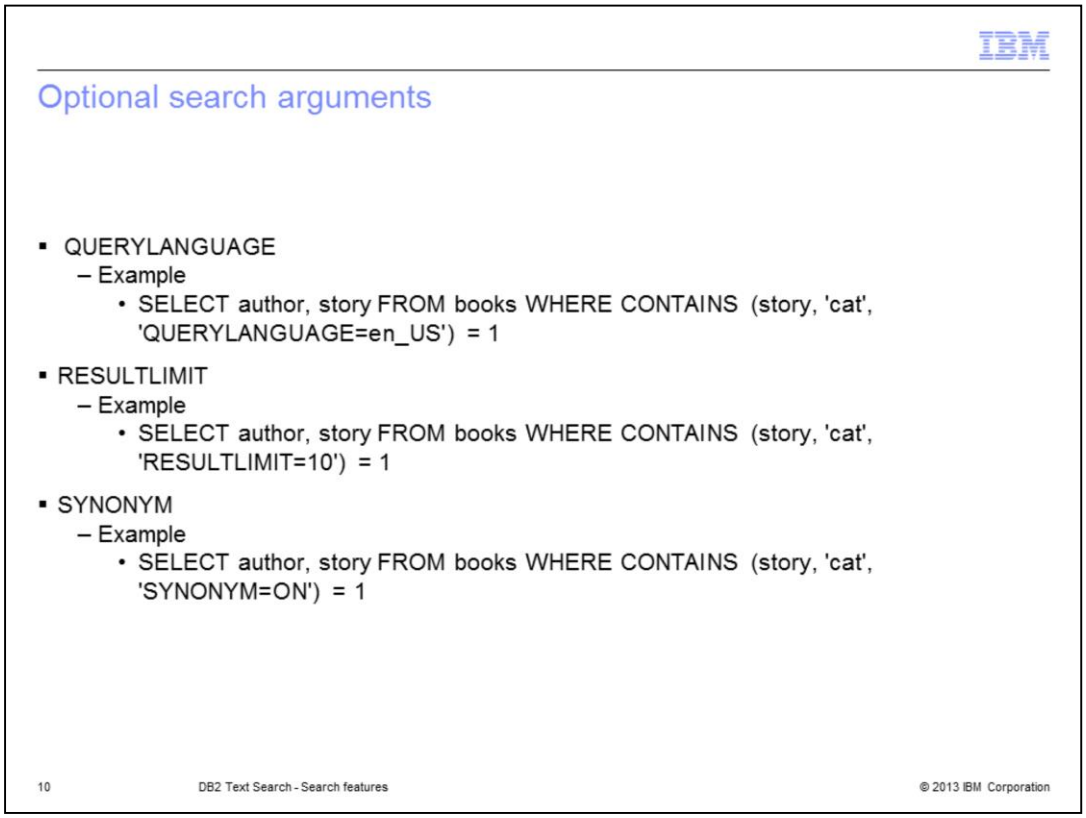
To give weight to or to boost score values for search terms, use the caret (^) character.



DB2 Text Search provides dictionary packs to support the linguistic processing of documents and queries. In addition, n-gram segmentation is supported for languages such as Chinese, Japanese, and Korean.

There are two processing options for Chinese, Japanese, and Korean languages. First, is the n-gram segmentation option which is the default and indexes overlapping pairs of characters. The second is the morphological segmentation option. It uses a dictionary specific to the language for identifying word boundaries.

The example displayed on this slide shows the difference between morphological and n-gram segmentation with a search for a meaningless Chinese word.

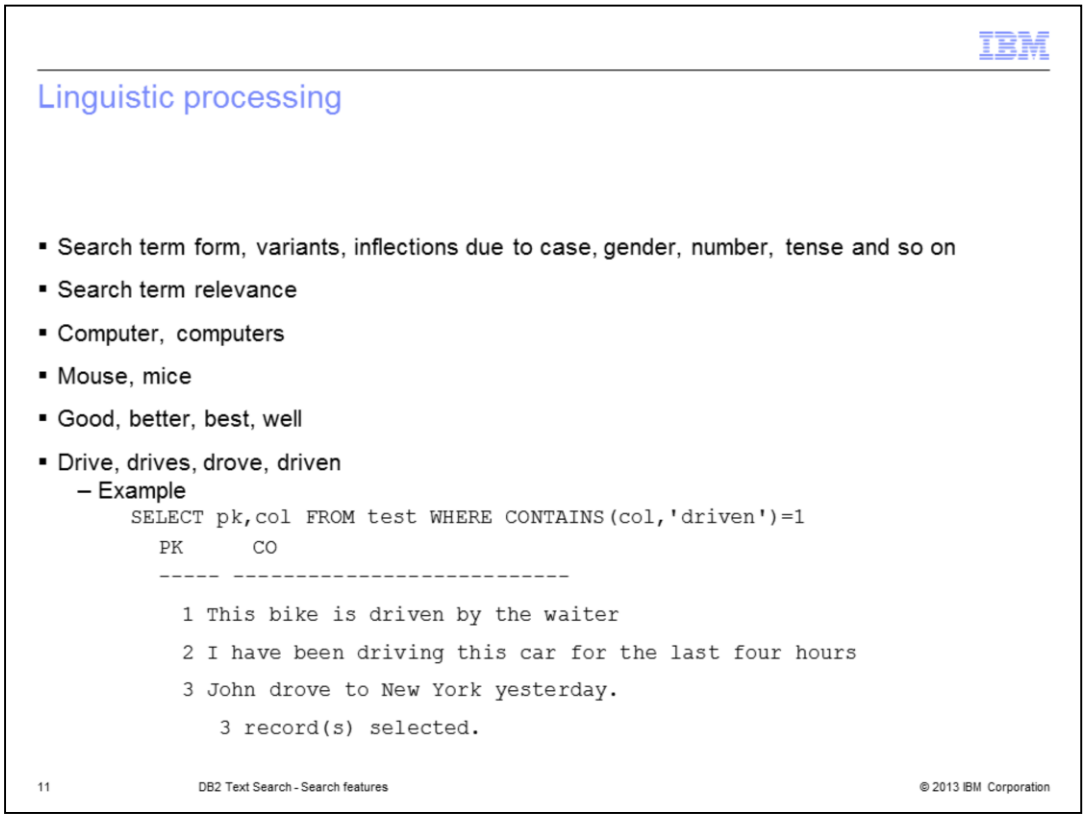


This slide displays the optional arguments that can be used in the search query.

QUERYLANGUAGE is used to explicitly specify the locale that DB2 Text Search uses to perform a search. If not provided, then the default is the locale of the text search index. In the example displayed on this slide, “en_US” locale is used.

RESULTLIMIT is used to limit the search results where a high count is anticipated. In the example displayed on this slide, the result is limited to 10 records.

SYNONYM is used to enable or disable the usage of a synonym dictionary related to the text search index. The default is OFF. Set it to ‘ON’ to search for synonyms of the query terms, thus, improving the result.

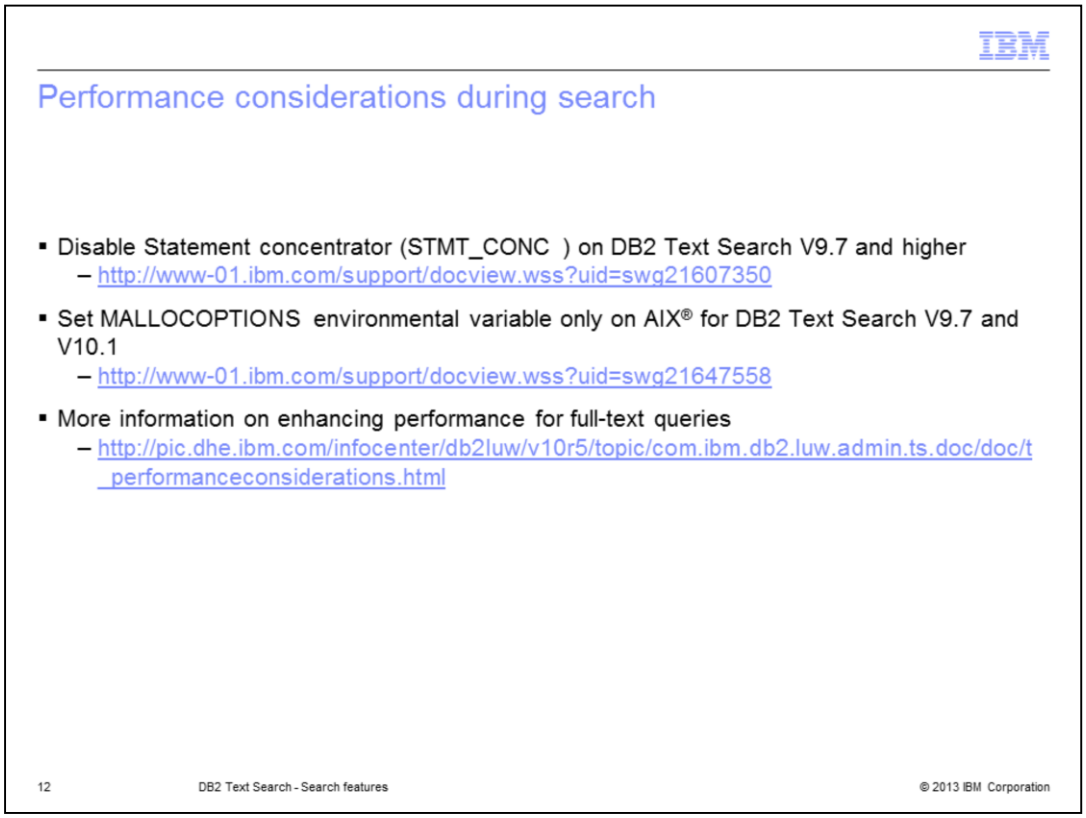


DB2 Text Search uses a dictionary based approach to perform linguistic processing to return only a relevant result set. Hence, depending on the language, linguistic variations of query terms can be returned.

For example, a search for good, as per linguistic processing, returns the documents that match good or better or best or well.

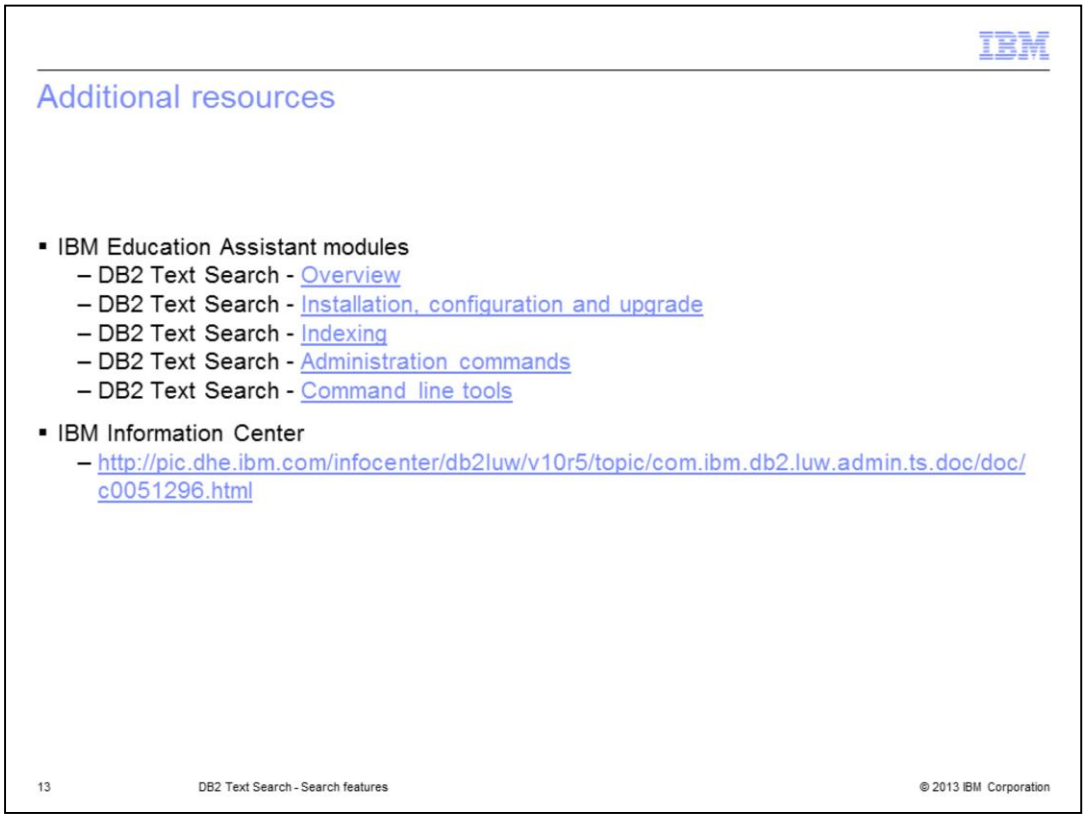
The example shows that the search for the term ‘driven’ returns the documents that contain driven, driving or drove.

For more details on linguistic processing, visit the Information Center.



It is expected that enabling the statement concentrator results in the generation of a sub-optimal plan for Text Search queries and hence, performance degradation is noticed. This parameter should be disabled for DB2 Text Search queries running on V9.7 or higher releases. For more details, go to the technote located at the link provided on this slide.

Another contributing factor for performance degradation can be due to a single memory heap allocation. This can be avoided by setting MALLOCOPTIONS environmental variable. It is applicable only on the AIX platform for V9.7 and V10.1. This variable setting is not required in V10.5 since DB2 automatically sets it during instance startup. For more details, go to the technote located at the link provided on this slide.



This slide displays additional IBM Education Assistant modules related to DB2 Text Search and it provides the link to the IBM Information Center.

Trademarks, disclaimer, and copyright information