

MICROSOFT
TRAINING
AND CERTIFICATION

Module 8: Querying Full-Text Indexes

Contents

Overview	1
Introduction to Microsoft Search Service	2
Microsoft Search Service Components	3
Getting Information About Full-Text Indexes	5
Writing Full-Text Queries	7
Recommended Practices	14
Lab A: Querying Full-Text Indexes	15
Review	21

Trainer Materials
for Microsoft Certified
Trainer Use Only



Information in this document is subject to change without notice. The names of companies, products, people, characters, and/or data mentioned herein are fictitious and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted. Complying with all applicable copyright laws is the responsibility of the user. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Microsoft Corporation. If, however, your only means of access is electronic, permission to print one copy is hereby granted.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2000 Microsoft Corporation. All rights reserved.

Microsoft, BackOffice, MS-DOS, PowerPoint, Visual Studio, Windows, Windows Media, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

The names of companies, products, people, characters, and/or data mentioned herein are fictitious and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Project Lead: Cheryl Hoople

Instructional Designer: Cheryl Hoople

Technical Lead: LeRoy Tuttle

Program Manager: LeRoy Tuttle

Graphic Artist: Kimberly Jackson (Independent Contractor)

Editing Manager: Lynette Skinner

Editor: Wendy Cleary

Editorial Contributor: Elizabeth Reese

Copy Editor: Bill Jones (S&T Consulting)

Production Manager: Miracle Davis

Production Coordinator: Jenny Boe

Production Tools Specialist: Julie Challenger

Production Support: Lori Walker (S&T Consulting)

Test Manager: Sid Benavente

Courseware Testing: Testing Testing 123

Classroom Automation: Lorrin Smith-Bates

Creative Director, Media/Sim Services: David Mahlmann

Web Development Lead: Lisa Pease

CD Build Specialist: Julie Challenger

Online Support: David Myka (S&T Consulting)

Localization Manager: Rick Terek

Operations Coordinator: John Williams

Manufacturing Support: Laura King; Kathy Hershey

Lead Product Manager, Release Management: Bo Galford

Lead Product Manager: Margo Crandall

Group Manager, Courseware Infrastructure: David Bramble

Group Product Manager, Content Development: Dean Murray

General Manager: Robert Stewart

Instructor Notes

Presentation:
45 Minutes

Lab:
60 Minutes

This module describes the Microsoft® Search service, how to access information about indexes on text fields, and how to query those full-text indexes. By the end of the module, students should be familiar with the capabilities of this service, but not expert in its use. At the end of this module, the student will be able to:

- Describe Microsoft Search service function and components.
- Get information about full-text indexes.
- Write full-text queries.

Materials and Preparation

Materials

To teach this module, you need the following materials:

- Microsoft PowerPoint® file 2017A_08.ppt.
- The C:\Moc\2071A\Demo\Ex_08.sql example file contains all of the example scripts from the module, unless otherwise noted in the module.

Preparation

To prepare for this module, you should:

- Read all of the materials.
- Complete the lab.

Trainer Materials
for Microsoft Certified
Trainer Use Only

Module Strategy

Use the following strategy to present this module:

- Introduction to Microsoft Search Service
Describe the concept and role of the Microsoft Search service in relation to Microsoft SQL Server™ 2000. Discuss the advantages of using the Microsoft Search service to index and query database tables in SQL Server.
- Microsoft Search Service Components
Describe the objects that the Microsoft Search service uses to implement full-text searching: full-text index and full-text catalog.
- Getting Information About Full-Text Indexes
Describe the various methods and information that you can use to gather information about full-text search by using full-text system stored procedures, Transact-SQL functions, and SQL Server Enterprise Manager. You can retrieve metadata and status information for each level—database, catalog, table, and column.
- Writing Full-Text Queries
Discuss how to write full-text queries by using Transact-SQL statements such as the CONTAINS and FREETEXT predicates and the CONSTAINSTABLE and FREETEXTTABLE functions. Point out when to use the predicates rather than the functions.

Customization Information

This section identifies the lab setup requirements for a module and the configuration changes that occur on student computers during the labs. This information is provided to assist you in replicating or customizing Microsoft Official Curriculum (MOC) courseware.

Important The lab in this module is dependent on the classroom configuration that is specified in the Customization Information section at the end of the *Classroom Setup Guide* for course 2071A, *Querying Microsoft SQL Server 2000 with Transact-SQL*.

Lab Setup

There are no lab setup requirements that affect replication or customization.

Lab Results

There are no configuration changes on student computers that affect replication or customization.

Overview

Slide Objective

To provide an overview of the module topics and objectives.

Lead-in

In this module, you will learn about Microsoft Search service and how to query full-text indexes.

- Introduction to Microsoft Search Service
- Microsoft Search Service Components
- Getting Information About Full-Text Indexes
- Writing Full-Text Queries

You can issue full-text queries against plain-text data in tables in Microsoft® SQL Server™ 2000, including words, phrases, and multiple forms of a word or phrase. You implement the full-text search capability in SQL Server by using Microsoft Search service.

At the end of this module, the student will be able to:

- Describe Microsoft Search service function and components.
- Get information about full-text indexes.
- Write full-text queries.

Introduction to Microsoft Search Service

Slide Objective

To describe the concept of Microsoft Search service in relation to SQL Server.

Lead-in

With Microsoft Search service, you now can query character-based data in tables by using full-text search.

■ Provides Text-based Searches in SQL Server

- Complex searches on unstructured text data for words and phrases
- More powerful than LIKE operator and wildcards

■ Available in Other Microsoft Products

- Indexing Service, Exchange 2000, Commerce Server

With Microsoft Search service, you now can query character-based data in tables by using full-text search. A *full-text query* searches for words, phrases, or multiple forms of a word or phrase in the character-based columns (**char**, **varchar**, **text**, **ntext**, **nchar**, or **nvarchar**).

Provides Text-based Searches in SQL Server 2000

SQL Server has always had the capability to retrieve text, based on pattern matching using the LIKE operator and wildcards.

Now, by using Microsoft Search service, you can create more complex searches. With a full-text query, you can perform a linguistic search of character data in tables. A linguistic search operates on words and phrases, allowing you to search for similar words or phrases and different forms of a word, or to target words that approximate one another.

Available in Other Microsoft Products

Microsoft Search service is also included with these products:

- Microsoft Indexing Service
- Microsoft Exchange 2000
- Microsoft Commerce Server

Microsoft Search Service Components

Slide Objective

To introduce the components of full-text search.

Lead-in

Microsoft Search service stores information about full-text data in structures outside of SQL Server.

■ Search Service Structures

- Full-text index
 - Keeps track of significant words used in a table
 - Requires unique key column or primary key
- Full-text catalog
 - Is a collection of full-text indexes

■ Microsoft Search Service Functions

- Indexing
- Querying

Microsoft Search service stores information about full-text data in structures outside of SQL Server.

Search Service Structures

SQL Server uses two structures to track columns that are full-text search enabled. These structures are used to communicate with Microsoft Search service for building indexes and processing queries.

Microsoft Search service uses two structures to implement full-text searching: full-text index and full-text catalog.

Full-Text Index

A *full-text index* is an index that keeps track of the significant words used in a table and where they are located. This index structure supports an efficient search for all items containing indexed words, and advanced search operations such as phrase searches and proximity searches.

The principal design requirement for full-text indexing, querying, and synchronization is the presence of a unique key column (or single-column primary key) on all tables that are registered for full-text search.

Note Full-text indexes differ from SQL Server indexes, which are controlled by the SQL Server database in which they are defined.

Full-Text Catalog

The full-text catalog is the location where full-text indexes reside. This is an NTFS file system directory that is accessible only by Microsoft Windows NT® Administrator and Microsoft Search service. The full-text indexes are organized into full-text catalogs. Typically, the full-text index data for an entire database is placed into a single full-text catalog. However, administrators have the flexibility to partition the full-text index data for a database across more than one full-text catalog. This is particularly useful if one or more of the tables being full-text indexed contains a large number of rows.

Search Service Functions

Microsoft Search service has two primary functions: building and maintaining full-text indexes, and using the indexes to process queries.

Indexing

Microsoft Search service builds and maintains full-text indexes for tables set up for full-text indexing. It goes through the rows of the table and extracts key words from the columns specified for full-text searching. These values are stored in operating system files and are organized into full-text catalogs. A table must have a unique index defined on it in order for you to build a full-text index on it.

Querying

Microsoft Search service processes full-text search queries. It determines which entries in the index meet the full-text selection criteria. For each entry that meets the selection criteria, it returns the identity of the row plus a ranking value to the SQL Server service, where this information is used to construct the query result set.

When processing a full-text query, the search engine returns to SQL Server the key values of the rows that match the search criteria.

Trainer Materials
for Microsoft Certified
Trainer Use Only

Getting Information About Full-Text Indexes

Slide Objective

To describe how to get information about full-text search.

Lead-in

You can gather information about full-text search.

- **Using System Stored Procedures**
 - `sp_help_fulltext_catalogs`
 - `sp_help_fulltext_tables`
 - `sp_help_fulltext_columns`
- **Using Transact-SQL Functions**
 - Use Transact-SQL functions to obtain information about full-text properties

```
USE northwind
SELECT
  DATABASEPROPERTY('Northwind', 'IsFullTextEnabled')
GO
```

You can gather information about full-text search by using full-text system stored procedures, Transact-SQL functions, and SQL Server Enterprise Manager. You can retrieve metadata and status information for each level—database, catalog, table, and column.

Using System Stored Procedures

You can use the full-text system stored procedures in the following table to obtain information about full-text indexes.

Stored procedure	Function
<code>sp_help_fulltext_catalogs</code>	Returns the ID, name, root directory, status, and number of full-text indexed tables for the specified full-text catalog
<code>sp_help_fulltext_tables</code>	Returns a list of tables that are enabled for full-text indexing
<code>sp_help_fulltext_columns</code>	Returns the columns that are enabled for full-text indexing

Using Transact-SQL Functions

You also can use Transact-SQL functions to obtain the values of full-text properties.

The following table lists frequently used property values that you can use to get information about full-text search.

Function	Property value	Description of Property
COLUMNPROPERTY	IsFullTextIndexed	Indicates that a column is enabled for full-text search
DATABASEPROPERTY	IsFullTextEnabled	Indicates that a database is enabled for full-text search
INDEXPROPERTY	IsFulltextKey	Identifies the index used by Microsoft Search service
OBJECTPROPERTY	TableHasActiveFulltextIndex	Indicates that a table is enabled for full-text search
FULLTEXTCATALOGPROPERTY	PopulateStatus	Returns the in-process state of a Microsoft Search catalog
	ItemCount	Returns the number of entries contained in a Microsoft Search catalog
	IndexSize	Returns the size of the full-text index, in megabytes
	UniqueKeyCount	Returns an approximate number of non-noise words that are able to be uniquely addressed in a Microsoft Search catalog
	LogSize	Returns the size of the last full-text index, in megabytes
FULLTEXTSERVICEPROPERTY	PopulateCompletionAge	Returns the most recent date and time at which an update was made to the referenced Microsoft Search catalog
	ResourceUsage	Specifies a relative operating system execution priority setting for the Microsoft Search service
	IsFullTextInstalled	Indicates a successful installation on an organization server running SQL Server

Example

The **IsFullTextEnabled** property is also used to determine whether full-text querying is enabled for the **Northwind** database.

```
USE northwind
SELECT DATABASEPROPERTY('Northwind', 'IsFullTextEnabled')
GO
```

If a value of 1 is returned, the **Northwind** database is enabled for full-text querying. A value of 0 indicates that the **Northwind** database is not enabled for full-text querying.

◆ Writing Full-Text Queries

Slide Objective

To describe how to write full-text queries.

Lead-in

With a full-text query, you can perform a linguistic search of character data in tables enabled for full-text search.

- **CONTAINS Predicate**
- **FREETEXT Predicate**
- **CONTAINSTABLE and FREETEXTTABLE Functions**
- **Using Full-Text Keywords**

Delivery Tip

The new Transact-SQL extensions follow the ISO SQL-3 functional methodology for full-text syntax extensions.

With a full-text query, you can perform a linguistic search of character data in tables enabled for full-text search.

Microsoft Search service supports additional search predicates and functions that extend the functionality of Transact-SQL. These syntax elements are the same as those used for Microsoft Search service in the OLE DB providers for Microsoft Index Server version 2.0 and Microsoft Site Server version 3.0.

The full-text Transact-SQL extensions follow the International Standards Organization (ISO) SQL-3 functional methodology for full-text syntax extensions. The syntax elements are:

- The CONTAINS predicate.
- The FREETEXT predicate.
- The CONTAINSTABLE function.
- The FREETEXTTABLE function.

Note Search predicates are Transact-SQL statements that act to restrict the result set of a query. An example is the search conditions applied in a WHERE clause of a SELECT statement.

CONTAINS Predicate

Slide Objective

To describe how to use the CONTAINS predicate.

Lead-in

You can use the CONTAINS predicate to search for a specific term.

- Inflectional Form of a Specific Word
- Word or Phrase Beginning with Specific Text
- Word or Phrase near Another Word or Phrase
- One or More Specific Words and Phrases
- Words or Phrases with Weighted Values

Example 1

```
SELECT plant_id, common_name, price
FROM plants
WHERE CONTAINS( *, ' "English Thyme" ' )
GO
```

You can use the CONTAINS predicate to search for a specific term.

Partial Syntax

The CONTAINS predicate uses functional notation in which the first parameter is the name of the column being searched, and the second parameter is a full-text search condition.

```
...WHERE CONTAINS ( {column}, '<contains_search_condition>' )
```

```
<contains_search_condition> ::=
  { | <generation_term> | <prefix_term> | <proximity_term>
    | <simple_term> | <weighted_term> }
```

Search Conditions Syntax Elements

The CONTAINS predicate supports complex syntax to search character-based columns for:

- Inflectional form of a specific word (*generation term*).
For example, you want to search for the inflectional form of the word drive. If various rows in the table include the words drive, drives, drove, driving, and driven, all would be in the result set, because each of these can be inflectionally generated from the word drive.
- A word or a phrase in which words begin with specified text (*prefix term*).
In a phrase, each word within the phrase is considered to be a prefix, such as the term auto tran* that matches both automatic transmission and automobile transducer.
- A word or phrase is near another word or phrase (*proximity term*).
For example, you want to find the rows in which the word ice is near the word hockey, or in which the phrase ice skating is near the phrase ice hockey.

- One or more specific words and phrases (*simple term*).
A word is one or more characters without spaces or punctuation. A valid phrase can consist of multiple words with spaces with or without punctuation between them. For example, ice is a word, and ice skating is a phrase. Words and phrases such as these are called *simple terms*. SQL Server discards noise words from the search criteria.
- Words or phrases with weighted values (*weighted term*).
You can rank the results of a search by specifying the weighted value of the words or phrases. Those query results that contain the higher-valued words or phrases are presented before others.

A CONTAINS predicate can combine several of these terms by using AND and OR, for example, to find all rows with latte and New York-style bagel in the same full-text enabled database column. Furthermore, terms can be negated by the use of AND NOT, for example bagel and not cream cheese.

Note A CONTAINS search is always case insensitive.

Example 1

The following query returns the **plant_id**, **common_name**, and **price** for all rows in which the phrase English Thyme is present in either of the full-text-enabled columns. An asterisk (*) used in place of a column name searches all full-text-enabled columns for the table.

```
SELECT plant_id, common_name, price
FROM plants
WHERE CONTAINS( *, ' "English Thyme" ' )
GO
```

Example 2

The following query returns such items as “Jean LeDuc has always loved ice hockey” and “Jean Leduc on Ice—Hockey at Its Best” from the **description** column.

```
SELECT article_id
FROM hockey_articles
WHERE CONTAINS(description, '"Jean LeDuc" AND "ice hockey"')
GO
```

FREETEXT Predicate

Slide Objective

To describe how to use the FREETEXT predicate.

Lead-in

When you use a FREETEXT predicate, you can enter any set of words or phrases, or even a complete sentence.

- Searches on Words, Phrases, and Sentences Extracted from Initial Query
- Less Precise Than the CONTAINS Predicate

```
SELECT *
FROM news_table
WHERE FREETEXT( description,
'The Fulton County Grand Jury said Friday an
investigation of Atlanta's recent primary
election produced no evidence that any
irregularities took place.')
GO
```

When you use the FREETEXT predicate, you can enter any set of words or phrases, or even a complete sentence. The full-text query engine examines this text, identifies all the significant words and noun phrases, and internally constructs a query with those terms. The FREETEXT predicate is less precise than the CONTAINS predicate.

Syntax

```
...WHERE FREETEXT ( {column | * }, 'free_text' )
```

Example

The following example uses a FREETEXT predicate against a column named **description**.

```
SELECT *
FROM news_table
WHERE FREETEXT( description,
'The Fulton County Grand Jury said Friday an investigation
of Atlanta's recent primary election produced no evidence that
any irregularities took place.')
GO
```

In the example, the search engine identifies the following words and noun phrases:

- Words—Fulton, county, grand, jury, Friday, investigation, Atlanta, recent, primary, election, produce, evidence, irregularities
- Phrases—Fulton county grand jury, primary election, grand jury, Atlanta's recent primary election

The words and phrases are combined internally into a query and weighted for proper ranking before the actual search is performed.

CONTAINSTABLE and FREETEXTTABLE Functions

Slide Objective

To introduce the CONTAINSTABLE and FREETEXTTABLE functions and compare them with the two previous keywords.

Lead-in

The CONTAINSTABLE and FREETEXTTABLE functions differ slightly from the CONTAINS and FREETEXT keywords.

- **Using CONTAINSTABLE and FREETEXTTABLE**
 - Return a table
 - Return column that contains full-text key values
 - *top_n_by_rank* argument
- **Choosing Which Search Condition or Function to Use**
 - Use the CONTAINS and FREETEXT predicates in any search condition
 - Use the CONTAINSTABLE and FREETEXTTABLE functions in the FROM clause

Delivery Tip

Point out that the CONTAINSTABLE and FREETEXTTABLE functions in effect create a derived table.

The CONTAINSTABLE and FREETEXTTABLE functions differ slightly from the CONTAINS and FREETEXT predicates. The CONTAINSTABLE and FREETEXTTABLE functions essentially create a derived table.

Using CONTAINSTABLE and FREETEXTTABLE

The CONTAINSTABLE and FREETEXTTABLE functions operate similarly to their counterpart search conditions, CONTAINS and FREETEXT. However, the functions return a table of zero, one, or more rows for those columns containing character-based data types. In addition, the CONTAINSTABLE and FREETEXTTABLE functions can be referenced in the FROM clause of a SELECT statement as though they were regular table names.

Partial Syntax

```
CONTAINSTABLE ( table , { column | * } , '< contains_search_condition >'
               [ , top_n_by_rank ] )
```

The CONTAINSTABLE function uses the same search conditions as the CONTAINS predicate.

Partial Syntax

```
FREETEXTTABLE ( table , { column | * } , 'freetext_string' [ , top_n_by_rank
                  ] )
```

When you work with these functions, consider the following facts and guidelines:

- CONTAINSTABLE and FREETEXTTABLE both return a table of zero, one, or more rows, so they must always be specified in a FROM clause.
- CONTAINSTABLE and FREETEXTTABLE also are used to specify selection criteria. The table returned has a column named **key** that contains full-text key values.

Each full-text-enabled table has a column whose values are guaranteed to be unique. The values returned in the **key** column are the full-text key values of the rows that match the selection criteria that are specified in the full-text search condition.

- The table produced by `CONTAINSTABLE` and `FREETEXTTABLE` has a column named **rank**, which contains values between 0 and 1,000. These values are used to relatively rank the rows returned according to how well they met the selection criteria, and have no comparison value outside the result set.

Tip Specify an integer for the *top_n_by_rank* argument to specify that only the *n* highest ranked matches, in descending order, be returned.

Choosing Which Full-Text Search Method to Use

Although the Transact-SQL statement used to specify the full-text search condition is the same in both the predicates and the functions, there are major differences in the way that you should use them.

- Use `CONTAINS` and `FREETEXT` predicates in any search condition (including a `WHERE` clause) of a `SELECT` statement.
- Use `CONTAINSTABLE` and `FREETEXTTABLE` functions in the `FROM` clause of a `SELECT` statement.

Queries that use the `CONTAINSTABLE` and `FREETEXTTABLE` functions are more complex than those that use the `CONTAINS` and `FREETEXT` keywords. You must explicitly join the qualifying rows returned by the functions with the rows in the original SQL Server table.

Trainer Materials
for Microsoft Certified
Trainer Use Only

Using Full-Text Keywords

Slide Objective

To describe how to combine full-text keywords with other Transact-SQL predicates.

Lead-in

You can use full-text keywords with other Transact-SQL statements.

```
USE northwind
SELECT Description
FROM Categories
WHERE CategoryName <> 'Seafood'
AND CONTAINS(Description, 'sauces AND seasonings ')
GO
```

CONTAINS and FREETEXT Predicates

```
USE pubs
SELECT T.title, P.pub_name
FROM publishers AS P
INNER JOIN titles AS T ON P.pub_id = I.pub_id
WHERE P.pub_id = (SELECT pub_id FROM pub_info WHERE CONTAINS
(pr_info, 'moonbeam AND ontario AND "flying saucer"'))
GO
```

CONTAINS Within a Subquery

You can combine the CONTAINS and FREETEXT predicates with any of the other Transact-SQL predicates, such as LIKE and BETWEEN. You also can use them in a subquery.

Example 1

This example searches for descriptions in which the category is not Seafood, and in which the description contains the word sauces and the word seasonings.

```
USE northwind
GO
SELECT Description
FROM Categories
WHERE CategoryName <> 'Seafood'
AND CONTAINS(Description, 'sauces AND seasonings ')
GO
```

Example 2

This example uses CONTAINS within a subquery. Using the **pubs** database, the query obtains the title value of all the books in the **titles** table for the publisher that is near the flying saucer landmark in Moonbeam, Ontario. (This information about the publisher is in the **pr_info** column in the **pub_info** table. Only one publisher matches this description.)

```
USE pubs
SELECT T.title, P.pub_name
FROM publishers AS P
INNER JOIN titles AS T ON P.pub_id = I.pub_id
WHERE P.pub_id = (
SELECT pub_id
FROM pub_info
WHERE CONTAINS (pr_info,
'moonbeam AND ontario AND "flying saucer"'))
)
GO
```

Recommended Practices

Slide Objective

To discuss recommended practices for querying full-text indexes.

Lead-in

These recommended practices should help you when querying full-text indexes.

✓ Use Full-Text Indexes on CHAR, VARCHAR, and TEXT Data Columns

✓ Use the Full-Text Index and Catalog Properties for Troubleshooting

✓ Use the *top_n_by_rank* Argument to Restrict Result Set Size

The following recommended practices should help you query full-text indexes:

- Use full-text indexes on tables with columns that contain textual data that you want to query, including those with CHAR, VARCHAR, and TEXT data types.
- Check the full-text index and catalog properties by using system functions and stored procedures to troubleshoot questionable query results or errors. Check that full-text search is enabled on the server, the database, and the table that you query.
- Use the *top_n_by_rank* argument when using CONTAINSTABLE and FREETEXTTABLE functions to restrict the number of returned rows.

Additional information on the following topics is available in SQL Server Books Online.

Topic	Search on
Using full-text search	“full-text query architecture” “full-text catalogs and indexes”
Querying full-text	“full-text query Transact-SQL components”
Using noise words	“indexing service”

Lab A: Querying Full-Text Indexes

Slide Objective

To introduce the lab.

Lead-in

In this lab, you will check the status of a full-text index and execute a query by using full-text indexes.



Explain the lab objectives.

Objectives

After completing this lab, you will be able to:

- Check the status of a full-text index.
- Write full-text queries.

Prerequisites

Before working on this lab, you must have:

- Answer files for this lab, which are located in C:\Moc\2071A\Labfiles\L08\Answers.

Lab Setup

To complete this lab, you must have:

- Executed the C:\Moc\2071A\Batches\2071A_R08.sql script file. This file is normally executed as part of the classroom setup. This script enables Microsoft Search service and creates a full-text index on the **Products.productname** column of the **Northwind** database.

For More Information

If you require help in executing files, search SQL Query Analyzer Help for “Execute a query”.

Other resources that you can use include:

- The **Northwind** database schema.
- SQL Server Books Online.

Scenario

The organization of the classroom is meant to simulate that of a worldwide trading firm named Northwind Traders. Its fictitious domain name is nwtraders.msft. The primary DNS server for nwtraders.msft is the instructor computer, which has an Internet Protocol (IP) address of 192.168.x.200 (where x is the assigned classroom number). The name of the instructor computer is London.

The following table provides the user name, computer name, and IP address for each student computer in the fictitious nwtraders.msft domain. Find the user name for your computer, and make a note of it.

User name	Computer name	IP address
SQLAdmin1	Vancouver	192.168.x.1
SQLAdmin2	Denver	192.168.x.2
SQLAdmin3	Perth	192.168.x.3
SQLAdmin4	Brisbane	192.168.x.4
SQLAdmin5	Lisbon	192.168.x.5
SQLAdmin6	Bonn	192.168.x.6
SQLAdmin7	Lima	192.168.x.7
SQLAdmin8	Santiago	192.168.x.8
SQLAdmin9	Bangalore	192.168.x.9
SQLAdmin10	Singapore	192.168.x.10
SQLAdmin11	Casablanca	192.168.x.11
SQLAdmin12	Tunis	192.168.x.12
SQLAdmin13	Acapulco	192.168.x.13
SQLAdmin14	Miami	192.168.x.14
SQLAdmin15	Auckland	192.168.x.15
SQLAdmin16	Suva	192.168.x.16
SQLAdmin17	Stockholm	192.168.x.17
SQLAdmin18	Moscow	192.168.x.18
SQLAdmin19	Caracas	192.168.x.19
SQLAdmin20	Montevideo	192.168.x.20
SQLAdmin21	Manila	192.168.x.21
SQLAdmin22	Tokyo	192.168.x.22
SQLAdmin23	Khartoum	192.168.x.23
SQLAdmin24	Nairobi	192.168.x.24

Estimated time to complete this lab: 60 minutes

Exercise 1

Obtaining Information About Full-Text Indexes

In this exercise, you will create a full-text catalog and full-text indexes so that you can use Microsoft Search service on one or more columns.

C:\Moc\2071A\Labfiles\L08\Answers contains completed scripts for this exercise.

► To use Transact-SQL to retrieve information about full-text indexes

In this procedure, you will set up Microsoft Search service on the **Northwind** database by using the Full-Text Indexing Wizard. `Answer_FullText1.sql` is a completed script for this procedure.

1. Log on to the **NWTraders** classroom domain by using the information in the following table.

Option	Value
User name	SQLAdminx (where <i>x</i> corresponds to your computer name as designated in the nwtraders.msft classroom domain)
Password	Password

2. Open SQL Query Analyzer and, if requested, log on to the (local) server with Microsoft Windows Authentication.

You have permission to log on to and administer SQL Server because you are logged as **SQLAdminx**, which is a member of the Windows 2000 local group, Administrators. All members of this group are automatically mapped to the SQL Server **sysadmin** role.

3. Execute the following statement to determine whether full-text searching capability is installed on the SQL Server computer:

```
USE northwind
SELECT fulltextserviceproperty('IsFullTextInstalled')
GO
```

4. Execute the following statement to determine whether full-text searching is enabled on the **Northwind** database:

```
USE northwind
SELECT databaseproperty('Northwind', 'IsFullTextEnabled')
GO
```

5. Execute the following system stored procedure to list the tables that have full-text search enabled:

```
USE northwind
EXEC sp_help_fulltext_tables
GO
```

6. Execute the following system stored procedure to list the columns that have full-text search indexes built on them:

```
USE northwind
EXEC sp_help_fulltext_columns
GO
```

Exercise 2

Writing Full-Text Queries

In this exercise, you will write and execute full-text queries against a full-text index in order to search for specific words and phrases in the **Northwind** database. C:\Moc\2071A\Labfiles\L08\Answers contains completed scripts for this exercise.

► **To check for the words St. Andrews near the word Scotland in the employee notes**

In this procedure, you will write and execute full-text queries with SELECT statements that use the CONTAINS and FREETEXT search functions. Answer_FullText2.sql is a completed script for this procedure.

1. Write a SELECT statement with the CONTAINS search function that selects the **LastName**, **Title**, **HireDate**, and **Notes** columns from the **Employees** table, where the **Employees.notes** column contains the word St. Andrews and is near the word Scotland.

```
USE northwind
SELECT LastName, Title, HireDate, Notes
FROM employees
WHERE CONTAINS(notes, '"St. Andrews" NEAR() "Scotland"')
GO
```

2. Execute the query to verify that it returns a single row.

► **To check for any form of the word sale in the employee notes**

In this procedure, you will check for any form of the word sale in the employee notes. Answer_FullText3.sql is a completed script for this procedure.

1. Write a SELECT statement with the CONTAINS search function that selects the **LastName**, **Title**, **HireDate**, and **Notes** columns from the **Employees** table, where the **Employees.notes** column contains any form of the word sale. Answer_FullText3.sql is a completed script for this procedure.

```
USE northwind
SELECT LastName, Title, HireDate, Notes
FROM employees
WHERE CONTAINS(notes, 'FORMSOF(INFLECTIONAL, "sale"')')
GO
```

2. Execute the query to verify that it returns four rows.

► **To check for the words cold and toast in the employee notes**

In this procedure, you will check for the words cold and toast in the employee notes. Answer_FullText4.sql is a completed script for this procedure.

1. Write and execute a SELECT statement with the FREETEXT search function that selects the **LastName**, **Title**, **HireDate**, and **Notes** columns from the **Employees** table, where the **Employees.notes** column contains the words cold and toast.

```
USE northwind
SELECT LastName, Title, HireDate, Notes
FROM employees
WHERE FREETEXT (notes, 'cold and toast')
GO
```

2. Execute the query to verify that it returns a single row.

► **To check for the words french and university in the employee notes of records that had orders shipped to London**

In this procedure, you will check for the words french and university in the employee notes that had order shipped to London. Answer_FullText5.sql is a completed script for this procedure.

1. Write a SELECT statement with the CONTAINS search function that selects the unique rows displaying the **LastName**, **Title**, and **HireDate** columns from the **Employees** table, where the **Employees.notes** column contains the words French and university, and the employee took orders that were destined for London.

```
USE northwind
SELECT DISTINCT LastName, Title, HireDate
FROM employees AS e
INNER JOIN Orders as o
ON e.EmployeeID = o.EmployeeID
WHERE ShipCity = 'London'
AND CONTAINS (notes, 'french and university')
GO
```

2. Execute the query to verify that it returns four rows.

► **To join a derived table that checks for the word sales near the word manager of records that had orders shipped to London**

In this procedure, you will join a derived table that checks for the word sales near the word manager of records that had orders shipped to London.

Answer_FullText6.sql is a completed script for this procedure, and the employee took orders that were destined for London.

1. Write a SELECT statement that displays **OrderID** and **EmployeeID** from the **Orders** table, ranking in descending order the best free-text search results for the word sales that is near the word manager in the **notes** column of the **Employees** table.

```
USE northwind
SELECT [RANK], OrderID, EmployeeID
FROM Orders AS o
INNER JOIN CONTAINSTABLE (Employees, notes,
                        "'sales' NEAR() 'manager'") AS e
ON o.EmployeeID = e.[KEY]
WHERE ShipCity = 'London'
ORDER BY [RANK] DESC
GO
```

2. Execute the query to verify that it returns six rows.

Trainer Materials
for Microsoft Certified
Trainer Use Only

Review

Slide Objective

To reinforce module objectives by reviewing key points.

Lead-in

The review questions cover some of the key concepts taught in the module.

- Introduction to Microsoft Search Service
- Microsoft Search Service Components
- Getting Information About Full-Text Indexes
- Writing Full-Text Queries

1. Why should you use the full-text search extensions instead of standard Transact-SQL syntax to query text data?

Standard Transact-SQL syntax only returns pattern matches on text fields and cannot take advantage of regular indexes. Full-text queries are indexed and optimized for word and proximity matches.

2. You issue a full-text search query and return an empty record set back for results. You know that the table contains data that should match. What should you do?

Determine the last time that the full-text index was updated by examining the PopulateCompletionAge property value. Also check the PopulateStatus and the ItemCount properties for the appropriate values.

3. You want to search the contents of the garden description column for any forms of “cultivate” that are near the word “roses.” Which keywords would you need to use in your SELECT statement to create this search?

CONTAINS, FORMSOF, and NEAR.

4. In your result set you would like to create a ranking. Which form of CONTAINS would you use?

CONTAINSTABLE

5. When you create a new project and load the database structure or insert a new table into an existing project, what table characteristic is required in order to define the semantic relationships for a table?

All tables require a defined primary key.

Trainer Materials
for Microsoft Certified
Trainer Use Only