

Re: FTI, Searching and other Filters

Source:

<http://www.tech-archive.net/Archive/SOL-Server/microsoft.public.sqlserver.fulltext/2006-12/msg00101.html>

- *From:* "Kyle Jedrusiak" <kjedrusiak@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 26 Dec 2006 13:24:13 -0500
-

I added a Office nvchar(8) column and populated it.

No matter what I specify for a search condition, it doesn't return what I'm after.

For a test I tried

```
select count(DocumentID) from Document where contains(*, 'Office02 and cobol')
```

'Office02' is in the new column, 'cobol' is in the image column.

If I search for them separately there is overlap so the data is correct.

Ideas?

"Lakusha" <Lakusha@xxxxxxxx> wrote in message <news:uWBdE0KJHHA.3552@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Kyle,

a) you can create an index on a view if you use VARCHAR(MAX) or VARBINARY(MAX) instead of TEXT or IMAGE (which are deprecated in sql 2005).

b) AFAIK, those "tokens" can be other columns from the same view/table that are also indexed in that FT catalog. You just specify CONTAINS(*,... Instead of CONTAINS(MyClobColumn,...

it should work.

"Kyle Jedrusiak" <kjedrusiak@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message <news:%23vIoXwFJHHA.536@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Looks like I'm stuck.

In order to create a Full Text Index on my view, the view has to have a

Re: FTI, Searching and other Filters

unique index.

SQL 2005 doesn't allow you to create an index on a view if the view contains an text, ntext, image or xml columns.

And mine does because we're storing the resume in an image column and the resume is what we're after.

We can't add any special tokens, the content is an image field.

Any other ideas?

"Simon Sabin" <SimonSabin@xxxxxxxxxxxxxxxx> wrote in message news:62959f1a36ede8c8edf7abdbb584@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hello Kyle,

The other option is to add the office ID to the content (If the content was editable i.e. text/html)

Then use a query like `containsstable(document, content, 'OFFICE2345 AND "SQL SERVER DBA "')`

If the content is editable this is by far the more manageable, and scalable.

We did the index view thing and it is just not a neat solution. The token thing is much easier.

Simon Sabin
SQL Server MVP
<http://sqlblogcasts.com/blogs/simons>

I think I found my answer...although a lot of work.

Remove the FTI from the table.

Add an OfficeID column to the table and populate it.

Partition the table by the OfficeID column.

Re: FTI, Searching and other Filters

Create an Indexed View for each OfficeID
(Open a new office, then add a new OfficeID
and a new Indexed View for
that
OfficeID.)
(Close an existing Office, migrate the
documents to a different
office, drop
the FTI for that View and drop that View)
Add a FTI to each of the Indexed Views.

Mod the application so it knows what how to
FT search one or more
Indexed Views and combine the results from
multiple views if needed.

"Kyle Jedrusiak"

<kjedrusiak@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote in
message

news:OitVbpuHHHA.1468@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

We have this table...

```
CREATE TABLE
[dbo].[Document](
[DocumentID] [int]
IDENTITY(1,1) NOT
NULL,
[HumanResourceID] [int]
NOT NULL,
[Name] [nvarchar](256)
COLLATE
SQL_Latin1_General_CP1_CI_AS
NOT NULL,
[Description]
[nvarchar](256) COLLATE
SQL_Latin1_General_CP1_CI_AS
NOT
NULL,
[ContentType] [nchar](4)
COLLATE
SQL_Latin1_General_CP1_CI_AS
NOT
NULL,
[Content] [image] NOT
NULL,
[DateEntered] [datetime]
NOT NULL,
[DateModified] [datetime]
```

Re: FTI, Searching and other Filters

```
NULL,  
[Version] [timestamp] NOT  
NULL,  
[EmployeeID] [int] NULL,  
CONSTRAINT  
[Resume_PK] PRIMARY  
KEY CLUSTERED  
(  
[ResumeID] ASC  
)WITH (PAD_INDEX =  
OFF, IGNORE_DUP_KEY  
= OFF, FILLFACTOR =  
90) ON  
[PRIMARY]  
) ON [PRIMARY]  
TEXTIMAGE_ON  
[PRIMARY]  
Content contains the bits  
that make up either Word or  
RTF documents.
```

We have FTI defined on
Content / ContentType /
DocumentID. Generally
FT searches are working.

The table contains over
130k documents.

Through our application we
are limiting their searches to
the top
1500 rank of any FT search.
(So as not to over-burden
our server.)

This works when the want
to search the table for
documents within the
entire company.

But what they would really
like is the top 1500 rank for
documents
within their office.

Is there a way to partition
the the table by an OfficeID
with some
what to pre-filter so the FT
search is only looking at

Re: FTI, Searching and other Filters

Documents from
one or more OfficeIDs?

TIA – Kyle!