

امکان استفاده‌ی همزمان قابلیت Full Text Search و اسناد XML ایی نیز در SQL Server پیش بینی شده‌است. به این ترتیب می‌توان متون این اسناد را ایندکس و جستجو کرد. در این حالت تگ‌های XML ایی و ویژگی‌ها، به صورت خودکار حذف شده و در نظر گرفته نمی‌شوند. Syntax استفاده از Full text search در اینجا با سایر حالات و ستون‌های متداول رابطه‌ای SQL Server تفاوتی ندارد. به علاوه امکان ترکیب آن با یک XQuery نیز میسر است. در این حالت، Full text search، ابتدا انجام شده و سپس با استفاده از XQuery می‌توان بر روی این نتایج، نودها، مسیرها و ویژگی‌های خاصی را جستجو کرد.

نحوه‌ی استفاده از Full Text Search بر روی ستون‌های XML ایی

برای آزمایش، ابتدا یک جدول جدید را که حاوی ستونی XML ایی است، ایجاد کرده و سپس چند سند XML را که حاوی متونی نسبتاً طولانی هستند، در آن ثبت می‌کنیم. ذکر CONSTRAINT در اینجا جهت دستور ایجاد ایندکس Full Text Search ضروری است.

```
CREATE TABLE ftsXML(
id INT IDENTITY PRIMARY KEY,
doc XML NULL
CONSTRAINT UQ_FTS_Id UNIQUE(id)
)
GO
INSERT ftsXML VALUES('
<book>
<title>Sample book title 1</title>
<author>Vahid</author>
<chapter ID="1">
<title>Chapter 1</title>
<content>
"The quick brown fox jumps over the lazy dog" is an English-language
pangram—a phrase that contains all of the letters of the English alphabet.
It has been used to test typewriters and computer keyboards, and in other
applications involving all of the letters in the English alphabet. Owing to its
brevity and coherence, it has become widely known.
</content>
</chapter>
<chapter ID="2">
<title>Chapter 2</title>
<content>
In publishing and graphic design, lorem ipsum is a placeholder text commonly used
to demonstrate the graphic elements of a document or visual presentation.
By replacing the distraction of meaningful content with filler text of scrambled
Latin it allows viewers to focus on graphical elements such as font, typography,
and layout.
</content>
</chapter>
</book>
')
```

```
INSERT ftsXML VALUES('
<book>
<title>Sample book title 2</title>
<author>Farid</author>
<chapter ID="1">
<title>Chapter 1</title>
<content>
The original passage began: Neque porro quisquam est qui dolorem ipsum quia dolor sit
amet consectetur adipisci velit
</content>
</chapter>
<chapter ID="2">
<title>Chapter 2</title>
<content>
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.
</content>
')
```

```
</chapter>
</book>
')
GO
```

سپس با استفاده از دستورات ذیل، Full text search را بر روی ستون doc جدول ایجاد شده، فعال می‌کنیم:

```
CREATE FULLTEXT CATALOG FT_CATALOG
GO
CREATE FULLTEXT INDEX ON ftsXML([doc])
KEY INDEX UQ_FTS_Id ON ([FT_CATALOG], FILEGROUP [PRIMARY])
GO
```

اکنون می‌توانیم با ترکیبی از امکانات Full Text Search و XQuery، از ستون doc، کوئری‌های پیشرفته و سریعی را تهیه کنیم.

راه اندازی سرویس Full Text Search

البته پیش از ادامه‌ی بحث به کنسول سرویس‌های ویندوز مراجعه کرده و مطمئن شوید که سرویس SQL Full-text Filter Daemon Launcher MSSQLSERVER در حال اجرا است. در غیراینصورت با خطای ذیل مواجه خواهید شد:

```
SQL Server encountered error 0x80070422 while communicating with full-text filter daemon host (FDHost) process.
```

اگر این سرویس در حال اجرا است و باز هم خطای فوق ظاهر شد، مجدداً به کنسول سرویس‌های ویندوز مراجعه کرد، در برگه‌ی خواص سرویس SQL Full-text Filter Daemon Launcher MSSQLSERVER، گزینه‌ی logon را یافته و آن‌را به local system account تغییر دهید و سپس سرویس را ری استارت کنید. پس از آن نیاز است دستور ذیل را نیز اجرا کنید:

```
sp_fulltext_service 'restart_all_fdhosts'
go
```

بعد از اینکار، بازسازی مجدد Full text search را فراموش نکنید. در این حالت در management studio، به بانک اطلاعاتی مورد نظر مراجعه کرده، نود Storage / Full Text Catalog را باز کنید. سپس بر روی FT_CATALOG ایجاد شده در ابتدای بحث کلیک راست کرده و از منوی ظاهر شده، گزینه‌ی Rebuild را انتخاب کنید. در غیراینصورت کوئری‌های ادامه‌ی بحث، خروجی خاصی را نمایش نخواهند داد.

استفاده از متد Contains

در ادامه، نحوه‌ی ترکیب امکانات Full text search و XQuery را ملاحظه می‌کنید:

```
-- استفاده از ایکس کوئری برای جستجو در نتایج حاصل
SELECT T.doc.value('/book/title)[1]', 'varchar(100)') AS title
FROM
-- استفاده از اف تی اس برای جستجو
(SELECT * FROM ftsXML
WHERE CONTAINS(doc, '"Quick Brown Fox "') AS T
```

ابتدا توسط متد Contains مرتبط به Full text search، ردیف‌های مورد نظر را یافته و سپس بر روی آن‌ها با استفاده از XQuery جستجوی دلخواهی را انجام می‌دهیم؛ از این جهت که Full text search تنها متون فیلدهای XML ایی را ایندکس می‌کند و نه تگ‌های آن‌ها را. خروجی کوئری فوق، 1 Sample book title است.

Full text search امکانات پیشرفته‌تری را نیز ارائه می‌دهد. برای مثال در ردیف‌های ثبت شده داریم fox jumps، اما در متن

ورودی عبارت جستجو، jumped را وارد کرده و به دنبال نزدیک‌ترین رکورد به آن خواهیم گشت:

```
SELECT T.doc.value('/book/title)[1]', 'varchar(100)') AS title
FROM
(SELECT * FROM ftsXML
WHERE CONTAINS(doc, 'FORMSOF (INFLECTIONAL , "Quick Brown Fox jumped")')) AS T
```

و یا دو کلمه‌ی نزدیک به هم را می‌توان جستجو کرد:

```
SELECT T.doc.value('/book/title)[1]', 'varchar(100)') AS title
FROM
(SELECT * FROM ftsXML
WHERE CONTAINS(doc, 'quick NEAR fox')) AS T
```

نکته‌ای در مورد متد Contains

هم Full text search و هم XQuery، هر دو دارای متدی به نام Contains هستند اما یکی نمی‌باشند.

```
SELECT doc.value('/book/title)[1]', 'varchar(100)') AS title
FROM ftsXML
WHERE doc.exist('/book/chapter/content[contains(., "Quick Brown Fox")]') = 1
```

در اینجا نحوه‌ی استفاده از متد contains مرتبط با XQuery را مشاهده می‌کنید. اگر این کوئری را اجرا کنید، نتیجه‌ای را دریافت نخواهید کرد. زیرا در ردیف‌ها داریم quick brown fox و نه Quick Brown Fox (حروف ابتدای کلمات، بزرگ نیستند). بنابراین متد contains مرتبط با XQuery یک جستجوی case sensitive را انجام می‌دهد.