

در دو قسمت قبل ابتدا سیستم FTS را نصب و فعال کردیم و سپس تعدادی رکورد را ثبت کرده، کاتالوگ‌های FTS، ایندکس‌ها و Stop words متناظری را ایجاد کردیم. در این قسمت قصد داریم از این اطلاعات ویژه، استفاده کرده و کوثری بگیریم. مواردی که بررسی خواهند شد اصطلاحاً Predicates نام داشته و شامل توابع مخصوصی مانند Contains و Freetext می‌شوند.

با استفاده از Contains predicate چه اطلاعاتی را می‌توان جستجو کرد؟

متد Contains مخصوص FTS، قابلیت یافتن کلمات و عبارات، تطابق کامل با عبارت در حال جستجو و یا حتی جستجوهای فازی را دارد. همچنین حالات مختلف صرفی یا inflectional یک کلمه را نیز می‌تواند جستجو کند (مانند jump, jumps و jumped). البته این مورد وابسته است به زبانی که در حین ایجاد ایندکس مشخص می‌شود. امکان یافتن کلماتی نزدیک و مشابه به کلماتی دیگر نیز پیش‌بینی شده‌است. پیشنوندها و پسوندها را نیز می‌توان جستجو کرد. امکان تعیین وزن و اهمیت کلمات در حال جستجو وجود دارند (برای مثال در این جستجوی خاص، کلمه‌ی ویژه اهمیت بیشتری نسبت به بقیه دارد). متد Contains امکان جستجوی Synonyms را نیز دارد. برای مثال یافتن رکوردهایی که معنایی مشابه need دارند اما دقیقاً حاوی کلمه‌ی need نیستند.

بررسی ریز جزئیات توانمندی‌های Contains predicate

1) جستجوی کلمات ساده

```
-- Simple term
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'data');
```

در این کوثری که بر روی جدول Documents قسمت قبل انجام می‌شود، به دنبال عین واژه‌ی در حال جستجو هستیم. باید دقت داشت که این نوع کوثری‌ها، حساس به حروف کوچک و بزرگ نیستند. همچنین عبارت وارد شده از نوع یونیکد است. به همین جهت برای جلوگیری از تغییر encoding رشته وارد شده (و تفسیر آن بر اساس Collation بانک اطلاعاتی)، یک N به ابتدای عبارت افزوده شده‌است.

2) جستجوی عبارات

```
-- Simple term - phrase
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'"data warehouse"');
```

اگر نیاز به یافتن عین عبارتی که از چند کلمه تشکیل شده‌است می‌باشد، نیاز است آن را با "" محصور کرد.

3) استفاده از عملگرهای منطقی مانند OR و AND

```
-- Simple terms with logical OR
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'data OR index');
```

در این کوثری نحوه‌ی استفاده از عملگر منطقی OR را مشاهده می‌کنید. و یا نحوه‌ی بکارگیری AND NOT در کوثری ذیل مشخص شده‌است:

```
-- Simple terms with logical AND NOT
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'data AND NOT mining');
```

در این کوئری به دنبال رکوردهایی هستیم که docexcerpt آن‌ها دارای کلمه‌ی data بوده، اما شامل mining نمی‌شوند. به علاوه با استفاده از پرانتزها می‌توان تقدم و تاخر عملگرهای منطقی را بهتر مشخص کرد:

```
-- Simple terms with many logical operators, order defined with parentheses
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'data OR (fact AND warehouse)');
```

4 جستجوی پیشوندها

```
-- Prefix
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'"add*");
```

در کوئری فوق به دنبال رکوردهایی هستیم که docexcerpt آن‌ها با کلمه‌ی add شروع می‌شوند. در این حالت نیز استفاده از "" اجباری است. اگر از "" استفاده نشود، FTS به دنبال تطابق عینی با عبارت وارد شده خواهد گشت.

5 جستجوهای Proximity

Proximity در اینجا به معنای یافتن واژه‌هایی هستند که نزدیک (از لحاظ تعداد فاصله بر حسب کلمات) به واژه‌ای دیگر می‌باشند.

```
-- Simple proximity
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'NEAR(problem, data)');
```

برای این منظور از واژه‌ی NEAR استفاده می‌شود؛ به همراه ذکر دو واژه‌ای که به دنبال آن‌ها هستیم. معنای کوئری فوق این است: رکوردهایی را پیدا کن که در آن در یک جایی از خلاصه سند، کلمه‌ی problem وجود دارد و در جایی دیگر از آن خلاصه‌ی سند، کلمه‌ی data.

همچنین می‌توان مشخص کرد که این نزدیک بودن دقیقاً به چه معنایی است:

```
-- Proximity with max distance 5 words
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'NEAR((problem, data),5)');

-- Proximity with max distance 1 word
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'NEAR((problem, data),1)');
```

در این کوئری‌ها اعداد 1 و 5، بیانگر فاصله‌ی بین دو کلمه‌ای هستند (فاصله بر اساس تعداد کلمه) که قرار است در نتایج جستجو حضور داشته باشند. مقدار پیش فرض آن Max است؛ یعنی در هر جایی از سند. همچنین می‌توان مشخص کرد که ترتیب جستجو باید دقیقاً بر اساس نحوه‌ی تعریف این کلمات در کوئری باشد:

```
-- Proximity with max distance and order
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'NEAR((problem, data),5, TRUE)');
GO
```

پارامتر آخر یا flag، به صورت پیش فرض false است. به این معنا که ترتیب این دو کلمه در جستجو اهمیتی ندارند.

6) جستجوی بر روی بیش از یک فیلد

در قسمت قبل، FULLTEXT INDEX انتهای بحث را بر روی دو فیلد docexcerpt و doccontent تهیه کردیم. اگر نیاز باشد تا جستجوی انجام شده هر دو فیلد را شامل شود می‌توان به نحو ذیل عمل کرد:

```
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS((docexcerpt,doccontent), N'data');
```

در این حالت تنها کافی است دو فیلد را داخل یک پرانتز قرار داد.

یک نکته: اگر تعداد ستون‌های ایندکس شده زیاد است و نیاز داریم تا بر روی تمام آن‌ها FTS انجام شود، تنها کافی است پارامتر اول متد Contains را * وارد کنیم. * در اینجا به معنای تمام ستون‌هایی است که در حین تشکیل FULLTEXT INDEX ذکر شده‌اند.

7) جستجوهای صرفی یا inflectional

FTS بر اساس زبان انتخابی، در حین تشکیل ایندکس‌های خاص خودش، یک سری آنالیزهای دستوری را نیز بر روی واژه‌ها انجام می‌دهد. همچنین امکان تعریف زبان مورد استفاده در حین استفاده از متد Contains نیز وجود دارد.

```
-- Inflectional forms
-- The next query does not return any rows
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'presentation');

-- The next query returns a row
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'FORMSOF(INFLECTIONAL, presentation)');
GO
```

در این مثال در کوئری اول به دنبال عین واژه‌ی وارد شده هستیم که با توجه به تنظیمات قسمت قبل و داده‌های موجود، خروجی را به همراه ندارد.

اکنون اگر کوئری دوم را که از FORMSOF جهت تعیین روش INFLECTIONAL استفاده کرده است، اجرا کنیم، به یک رکورد خواهیم رسید که در آن جمع واژه‌ی presentation وجود دارد.

8) جستجو برای یافتن متشابهات

برای نمونه اگر SQL Server 2012 بر روی سیستم شما نصب باشد، محل نصب واژه‌نامه‌های Synonyms یا واژه‌هایی همانند از لحاظ معنایی را در مسیر زیر می‌توانید مشاهده کنید:

```
C:\...\MSSQL11.MSSQLSERVER\MSSQL\FTData
```

این‌ها یک سری فایل XML هستند با ساختار ذیل:

```
<XML ID="Microsoft Search Thesaurus">
  <thesaurus xmlns="x-schema:tsSchema.xml">
    <diacritics_sensitive>0</diacritics_sensitive>
    <expansion>
      <sub>Internet Explorer</sub>
      <sub>IE</sub>
      <sub>IE5</sub>
    </expansion>
    <replacement>
      <pat>NT5</pat>
      <pat>W2K</pat>
```

```

        <sub>Windows 2000</sub>
    </replacement>
    <expansion>
        <sub>run</sub>
        <sub>jog</sub>
    </expansion>
    <expansion>
        <sub>need</sub>
        <sub>necessity</sub>
    </expansion>
</thesaurus>
</XML>

```

در اینجا diacritics_sensitive به معنای حساسیت به لهجه است که به صورت پیش فرض برای تمام زبان‌ها خاموش است. سپس یک سری expansion و replacement را مشاهده می‌کنید.

فایل tsenu.xml به صورت پیش فرض برای زبان انگلیسی آمریکایی مورد استفاده قرار می‌گیرد. اگر محتویات آن را برای مثال با محتویات XML ایی فوق جایگزین کنید (در حین ذخیره باید دقت داشت که encoding فایل نیاز است Unicode باشد)، سپس باید SQL Server را از این تغییر نیز مطلع نمائیم:

```

-- Load the US English file
EXEC sys.sp_fulltext_load_thesaurus_file 1033;
GO

```

عدد 1033، عدد استاندارد زبان US EN است.

البته اگر اینکار را انجام ندهیم، به صورت خودکار، اولین کوثری که از THESAURUS انگلیسی استفاده می‌کند، سبب بارگذاری آن خواهد شد.

```

-- Synonyms
-- The next query does not return any rows
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'need');

-- The next query returns a row
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(docexcerpt, N'FORMSOF(THESAURUS, need)');
GO

```

در اولین مثال به دنبال عین واژه‌ی need در رکوردهای موجود هستیم که خروجی را بر نمی‌گرداند.

در ادامه اگر کوثری دوم را که از FORMSOF جهت تعیین روش THESAURUS استفاده کرده است، اجرا کنیم، به یک رکورد خواهیم رسید که در آن واژه‌ی necessity به کمک محتویات فایل tsenu.xml که پیشتر تهیه کردیم، بجای need وجود دارد.

9 جستجو بر روی خواص و متادیتای فایل‌ها

```

-- Document properties
SELECT id, title, docexcerpt
FROM dbo.Documents
WHERE CONTAINS(PROPERTY(doccontent, N'Authors'), N'Test');

```

در اینجا نحوه‌ی جستجوی خواص فایل‌های docx ذخیره شده در قسمت قبل را مشاهده می‌کنید که شامل ذکر PROPERTY و ستون FTS مورد نظر است، به همراه نام خاصیت و عبارت جستجو.

کار با FREETEXT

```

-- FREETEXT
SELECT *
FROM dbo.Documents
WHERE FREETEXT(docexcerpt, N'data presentation need');

```

FREETEXT عموماً ردیف‌های بیشتری را نسبت به Contains بر می‌گرداند؛ چون جستجوی عمومی‌تری را انجام می‌دهد. در اینجا جستجو بر روی معنای عبارات انجام می‌شود و نه صرفاً یافتن عباراتی دقیقاً همانند عبارت در حال جستجو. در اینجا مباحث Inflectional و Synonyms ایی که پیشتر یاد شد، به صورت خودکار اعمال می‌شوند. در کوئری فوق، کلیه رکوردهایی که با سه کلمه‌ی وارد شده (به صورت مجزا) به نحوی تطابق داشته باشند (تطابق کامل یا بر اساس تطابق‌های معنایی یا دستوری) باز گردانده خواهند شد.

نظرات خوانندگان

نویسنده: میلاد رسولی ا
تاریخ: ۲۱:۱۷ ۱۳۹۴/۰۱/۰۸

بنده در حال ساخت جستجویی برای وب سایتی هستم. این جستجو بر روی جداول کتاب، نویسنده، مترجم و انتشارات انجام میشه و در صورتی که کاربر قسمتی از نام کتاب و نام نویسنده را وارد کند جستجو بر روی این دو فیلد که از دو جدول متفاوت هستند انجام می‌شود.

مشکل اینجاست که از آنجایی که دستوارت FTS بر روی یک جدول عمل می‌کنند و با توجه به پیچیدگی جستجو، شما چه راهی را برای کوئری گرفتن از چندین جدول (که ممکن است یک کتاب چند نویسنده هم داشته باشد) پیشنهاد می‌کنید.

بنده در حال حاضر تمام این جداول را در یک View قرار داده و فیلدهای چندمقداری را با Concat بوسیله " " در یک فیلد جای داده‌ام.

ممنون از راهنماییتون

نویسنده: وحید نصیری
تاریخ: ۲۲:۱۶ ۱۳۹۴/۰۱/۰۸

دو متد [CONTAINSTABLE](#) و [FREETEXTTABLE](#) امکان join را نیز میسر می‌کنند.

```
SELECT b.Name, a.Name, bkt.[Rank] + akt.[Rank]/2 AS [Rank]
FROM Book b
INNER JOIN Author a ON b.AuthorID = a.AuthorID
INNER JOIN FREETEXTTABLE(Book, Name, @criteria) bkt ON b.ContentID = bkt.[Key]
LEFT JOIN FREETEXTTABLE(Author, Name, @criteria) akt ON a.AuthorID = akt.[Key]
ORDER BY [Rank] DESC
```