

組み込み型全文検索

現場で働く作業員は、日々のテキスト・ドキュメントやメッセージなどのやり取りにおいて、半構造化データを扱う必要性がますます高まっています。現在の SQL Anywhere (Ver.11 以降)には、組み込み型全文検索が装備されているため、アドオンは必要ありません。全文検索を使用すれば、テーブルのローをスキャンしたり、どのカラムに用語が保存されているかを調べたりする必要なく、データベース内で用語 (単語) の全インスタンスをすばやく検索できます。

組み込み型全文検索の例

この例では、SQL Anywhere Ver.11 で全文検索をセットアップする方法を示し、検索で新しい CONTAINS 句がどのように使用されるかについて説明します。

1. Interactive SQL を起動し、ODBC から SQL Anywhere 11 Demo データベースに接続します。

Windows の場合、コマンドプロンプトを開きます。

Linux の場合、Terminal を開き、SQL Anywhere のサンプルがインストールされているかどうかをまだ確認していない場合は、`$SQLANY11/samples/sample_config32.sh` のファイルを実行して確認します。

以下のコマンドを実行します。

```
dbisql -c "dsn=SQL Anywhere 11 Demo"
```

2. Interactive SQL で ([F5] を押して) 以下の文を実行することにより、*MyTxtConfig* というテキスト構成オブジェクトを作成します。

```
CREATE TEXT CONFIGURATION myTxtConfig FROM  
default_char;
```

テンプレートとして使用するテキスト構成オブジェクトを指定する FROM 句を必ず含めてください。

3. テキスト・オブジェクトをカスタマイズします。ストップ・リスト (インデックス化するときに無視する用語) を追加し、最小長に 4、最大長に 30 を設定します。これらを設定するには、以下の 3 つの ALTER TEXT CONFIGURATION 文を使用します ([F5] を押して実行)。

```
ALTER TEXT CONFIGURATION myTxtConfig  
  STOPLIST 'because about therefore only';  
ALTER TEXT CONFIGURATION myTxtConfig  
  MINIMUM TERM LENGTH 4;
```

```
ALTER TEXT CONFIGURATION myTxtConfig
MAXIMUM TERM LENGTH 30;
```

4. Products テーブルの Name および Description カラムに基づいてテキスト構成オブジェクトの *myTxtConfig* を参照するテキスト・インデックスを作成します。この例では、テーブルのデータが頻繁に変更されないため、リフレッシュ間隔は 24 時間で十分であると想定します。以下の文を実行します。

```
CREATE TEXT INDEX myTxtIdx ON Products ( Name,
Description )
CONFIGURATION myTxtConfig
AUTO REFRESH EVERY 24 HOURS;
```

5. テキスト・インデックスを初期化します ([F5] を押して実行)。

```
REFRESH TEXT INDEX myTxtIdx ON Products;
```

6. いくつかの異なるクエリを使用して、テキスト・インデックスをテストします。1 つ目のクエリは、説明に *cotton* または *caps* が含まれているすべての製品を返します。

```
SELECT ID, Name, Description, Quantity
FROM Products
WHERE CONTAINS ( Products.Description, 'cotton | caps' );
```

Results				
	ID	Name	Description	Quantity
1	700	Shorts	Cotton Shorts	80
2	400	Baseball Cap	Cotton Cap	112

- 2 つ目のクエリは、クエリの選択リストにおける全文検索の使用方法を示します。

```
SELECT IF CONTAINS ( Description, 'cotton' )
THEN 1
ELSE 0
ENDIF AS Results
FROM Products;
```

Results	
	Results
1	0
2	0
3	0
4	1
5	0
6	0
7	0
8	0
9	0
10	1

3 つ目のクエリは、結果ローが目的の検索文字列にどのくらい近いかを表すスコアが返される方法や、その情報を使用して結果セットが順序付けられる方法を示します。

```
SELECT Products.Description, ct.*
FROM Products CONTAINS ( Products.Description,
'caps OR cotton' ) ct
ORDER BY ct.score DESC;
```

Results		
	Description	score
1	Cotton Cap	2.297861303197614
2	Cotton Shorts	2.046015331262012

結論

複数のカラムおよび句にわたって複数の用語を検索できることに加え、類似検索（例については、オンライン・ドキュメントを参照）も実行できるため、全文検索はテキスト・フィールドでの単一用語の検索パフォーマンスを著しく向上させます。以下のグラフは、標準的な WHERE 句 (LIKE) を使用する場合と全文検索 (CONTAINS 文) を使用する場合の、LONG VARCHAR フィールドの簡単な検索の実行時間を比較したものです。この例では、500MB のデータベースが使用されました。全文検索を使用した場合の実行時間 (2 秒) の方が使用しない場合 (20 秒) よりもはるかに高速であることがわかります。

