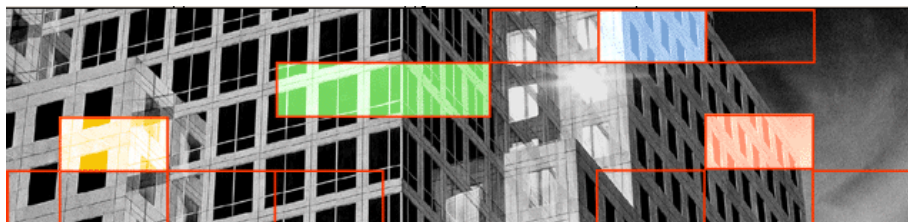


## Fulltextové vyhledávání v MS SQL Serveru

Zápočtová práce do předmětu Databázové systémy 2



# Microsoft SQL Server

Velké množství digitálních informací je uloženo v nestrukturovaném textu, buď v čistém textu, nebo ve formátovaných dokumentech. Vyhledávání určitých informací v nich je velmi důležité v mnoha aspektech používání IT. Ať už je to vyhledávání přes Google, vyhledávání frází ve slovnících nebo prosté hledání slova v některém z adresářů, fulltextové vyhledávání je nejefektivnější způsob jak se k daným informacím dostat. Tuto možnost nabízí (i když ne v takovém rozsahu jako Oracle) také MS SQL a Microsoft produkty obecně.

Samotné Microsoft fulltextové vyhledávání funguje pro

- Fulltextové dotazy na čistá textová data
- Kombinování fulltextového vyhledávání s tradičními vyhledávacími algoritmy
- Hledání v indexovaných tabulkách
- Procentuální shody ve vyhledávání s danými výrazy na základě tzv. „Ranku“
- Prohledávání strukturovaných dat typu HTML, Office dokumentů
- Vytváření speciálních filtrů pro další typy souborů (PDF, TIFF)

Základním úkolem jazykových modulů pro fulltextové vyhledávání je najít fráze, které uživatel hledá a to buď přesně, nebo identifikovat syntakticky různé tvary, jimž odpovídá tentýž faktický význam a ty pak najít.

Nejdůležitějším případem je identifikace různých tvarů ohebných slovních druhů, zejména podstatných jmen, přídavných jmen a sloves. U jazyků se složitým tvaroslovím, jako jsou právě slovanské jazyky, kde např. podstatná jména mají dva časy krát sedm pádů jde o velmi důležitou funkci, bez níž je fulltextové hledání přinejlepším neúplné. Služba, která se stará o fulltextové vyhledávání v systémech Windows se jmenuje Microsoft Search Service.

### Microsoft Search Service

Microsoft Search service je fulltextová a vyhledávací služba (dále jen MSS)

MSS má dvě úlohy

- Podpora indexování – vytváří fulltextové katalogy a indexy založené pro databázi. Při definici katalogů vytváří indexy pro každý z nich.

- Podpora dotazování – zjišťuje, které výrazy splňují daná vyhledávací kritéria. Pro každý dotaz vrací řádky v tabulce spolu s Rankem – což je jakási velikost shody s kritérii v dotazu.

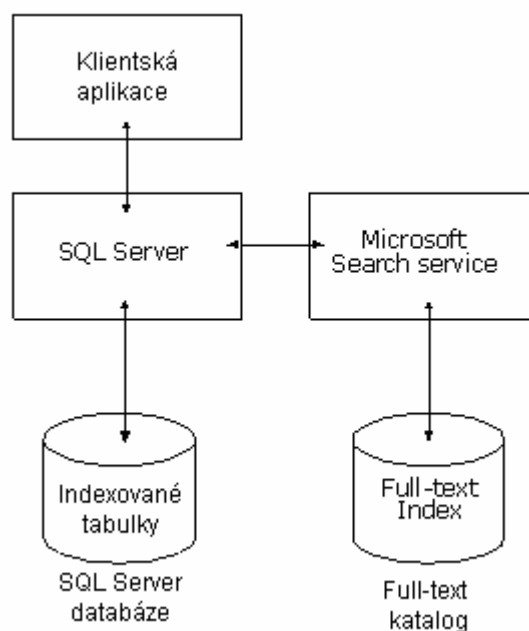
Samotná služba běží implicitně na Windows NT, 2000, XP a Serverech. Na starších systémech je možné ji zprovoznit dodatečnou instalací balíčků.

MSS běží s právy administrátora, jehož práva si během instalace sama přiřadí.

Fulltextové katalogy a indexy nejsou uloženy přímo v databázi, ale ve speciálních souborech, spravovaných službou MSS. Přístup k nim mají pouze MSS a systémový administrátor.

Z obrázku je vidět že samotná databáze a MSS pracují zvlášť. Jejich spolupráci zajišťuje fulltext provider.

Fulltextové katalogy, indexy a podpory hledání využívá MSS pouze na tabulky v databázi. Pokud je třeba vyhledávat v klasických souborech, je použit Index Service, který poskytuje stejné funkce jako MSS ale právě na klasické soubory. Index Service obsahuje také OLE DB ovladače, díky nimž je možné ho propojit s databázemi. Následně je pak možné v T-SQL dotazech (Transact - SQL) automaticky kombinovat vyhledávání v databázi s prohledáváním skutečných souborů.



## 1. Princip činnosti

Činnost fulltextového systému lze rozdělit na dvě části: vytváření indexu a vyhledávání v indexu.

### 1.1. Vytváření fulltextového indexu

Při vytváření fulltextového indexu dochází postupně k těmto krokům:

- Filtrování – během tohoto kroku dochází k extrakci čistého textu z indexovaného dokumentu. Tento krok není vždy nutný (například pokud jde o textové pole v databázi nebo TXT soubor). Jindy je naopak nezbytný, neboť indexovaná data mohou obsahovat řadu netextových informací (jako např. soubory MS Office). Standardně je dodáván *IFilter* modul pro soubory MS Office, TXT soubory, XML a HTML soubory. Od třetích stran lze získat např. filtry pro PDF nebo TIFF dokumenty. Filtrování je jazykově nezávislé.
- Určení jazyka – zároveň s filtrací dochází k určení jazyka dokumentu. Pokud existují metadata definující tento jazyk (např. ve formátech MS Office, v hlavičkách HTML dokumentů, v definici fulltextového vyhledávání v SQL tabulkách), použijí se tato metadata. Pokud není žádný takový údaj k dispozici, je jako použitý jazyk určen vybraný systémové místní nastavení (*System Default Locale*) počítače, který indexaci provádí.

- Rozdělení na slova (*word breaking*) – během této fáze je neformátovaný text rozdělen na jednotlivá slova a výrazy. Tato činnost je jazykově závislá. U evropských jazyků je dělení na slova dosti podobné, ale přesto se vyskytují rozdíly, např. „č“ je v češtině a slovenštině považováno za řádný znak, zatímco v angličtině je považováno za speciální symbol, který není součástí slova. Zároveň dochází ke korektnímu rozeznání a zpracování složených slov (slova obsahující spojovník nebo lomítko) a zkratk. V této fázi též probíhá normalizace (unifikace) číselných údajů (čísla, data, časy), což umožňuje např. nalezení data bez ohledu na použitý formát jeho výskytu v textu (například 2.2.99, 02.02.1999 a 2.2.1999).
- Odstranění nevýznamných slov (*noise words*) – pro každý jazyk je k dispozici seznam nevýznamných slov (pro češtinu např. a, ale, tak, tedy), která jsou odstraněna z důvodu menší velikosti indexu a lepší výkonnosti. Tato činnost je jazykově závislá.
- Uložení do fulltextového indexu – zjednodušeně si lze index představit jako seznam slov, kde u každého slova je uvedeno v kterém dokumentu se vyskytuje a na kterých pozicích (aby bylo možné použít při vyhledávání operátor NEAR). Jde o jazykově nezávislou činnost.

## 1.2. Vyhledávání

Při dotazování do indexu se využívá již hotový fulltextový slovník. Při dotazech lze využívat jednoduché i pokročilejší operátory. Při vyhledávání hraje důležitou roli takzvaný *stemmer*, jehož úkolem je rozložit dotaz podle všech možných tvarů daného slova. Například pokud zadáme dotaz:

*Havlíček AND Němcová*

lze si výsledný dotaz představit jako

*(Havlíček OR Havlíčka OR Havlíčkem OR ...) AND (Němcová OR Němcovou OR ...)*

Z toho je jasně patrné, že správné určení jazyka při pokládání dotazu je klíčové pro úspěch vyhledávání. Některé produkty využívající fulltextové vyhledávání nabízejí možnost volby jazyka, jiné používají nějakou formu výchozího nastavení. "

Jazykové moduly pro češtinu/slovenštinu pracují na bázi rozsáhlých slovníků. Z toho plyne, že pro slova obsažená ve slovnících je generování odvozených tvarů (ohybů slova) přesné a úplné, a to i v případech, kdy dochází ke změně základu slova pro některé tvary slova (nepravidelné skloňování nebo časování). Ohyby jsou generované v rámci jednoho slovního druhu. Pro podstatná jména jsou generovány všechny tvary v jednotném i množném čísle, pro slovesa tvary přítomného času, minulého času, rozkazovacího způsobu, neurčitku a přechodníky, pro přídavná jména a zájmena všechny tvary v jednotném a množném čísle pro všechny rody. Slovesná podstatná jména, přivlastňovací přídavná jména a přídavná jména jsou zpracovávána jako samostatná slova, tzn. negenerují se slovní druhy, ze kterých jsou odvozené (přesvědčení a přesvědčit jsou považována za různá slova). Záporné tvary slov vytvářené pomocí ne- jsou zpracovávány odděleně od kladných tvarů (slova nevědět a vědět jsou považována za různá). Důležitou vlastností jazykových modulů je schopnost generovat odvozené tvary (ohyby) i pro slova, která se ve slovnících nenachází (především příjmení, názvy, nově vznikající/převzatá a slangová slova). Pro generování ohybů neznámých slov jsou použity algoritmy, které s vysokou přesností určí slovní druh a další gramatické atributy neznámého slova (rod, číslo, čas, ...) a v závislosti na těchto informacích je generována příslušná množina odvozených tvarů slova.

## 2. Dotazování

MS SQL Server poskytuje efektivní podporu pro hledání slov v textových datech. Fulltextový index ukládá informace o jednotlivých slovech a jejich umístění v tabulkách a následně jsou tyto informace používány pro rychlé vyhledávání frází a jejich kombinací v těchto tabulkách.

Fulltextové indexy jsou obsaženy ve fulltextových katalozích. Každá databáze může obsahovat mnoho těchto katalogů, katalog však nesmí náležet do více než jedné databáze a každý katalog může obsahovat fulltextové indexy pro více tabulek. Tabulka smí mít pouze jeden fulltextový index, takže každá indexovaná tabulka náleží vždy pouze do jednoho fulltextového katalogu.

Samotné indexy musí být definovány na základní tabulku, ne na pohled, systémovou tabulku nebo na dočasnou tabulku. K definování indexu je třeba definovat unikátní sloupec a jeden nebo více textových sloupců. Každý index pak obsahuje hodnoty určitých slov, které jsou s ním spojeny a umístění slov v textu.

Některé řetězce nemohou být ukládány do klasických textových sloupců. V MS SQL je pro tyto soubory (jako jsou například Office dokumenty) vytvořen speciální typ image. Office i jiné dokumenty totiž obsahují mnoho znaků, které nejsou pro ukládání do klasických text sloupců vhodné. Naproti tomu image umožňuje ukládat jakákoliv data. Při vyhledávání je pak u těchto sloupců použit filtr, který dokáže z daných dat vytáhnout pouze čistý text. Filtr je možné zvolit explicitně, nebo nechat SQL Server zvolit filtr podle přípony souboru.

Fulltextové indexování je komponenta, která umožňuje dvě možnosti T-SQL dotazování při hledání daných frází a to:

- CONTAINS
- FREETEXT

Transact-SQL také obsahuje dvě funkce, které vrací množinu řádků, které odpovídají vyhledávacím kritériím a to :

- CONTAINSTABLE
- FREETEXTTABLE

Ve chvíli, kdy je spuštěn fulltextový dotaz, je přeposlán do MSS, které projde své vnitřní struktury (indexy) a vrátí klíče a ranky zpět SQL Serveru. Tento proces probíhá na pozadí a není možné prohlížet jeho výsledky, když je použit dotaz CONTAINS nebo FREETEXT, ale jestliže je spuštěn dotaz CONTAINSTABLE nebo FREETEXTTABLE, můžeme se do k daným hodnotám dostat. Samotné dotazy CONTAINS a FREETEXT vlastně nedělají nic jiného, než že zavolají CONTAINSTABLE nebo FREETEXT a výsledky propojí s jejich tabulkou v databázi a vrátí výsledek.

### 2.1. CONTAINS

Je určeno k vyhledávání sloupců obsahujících texty s naprostou shodou, nebo fuzzy (méně přesnou) shodou, jsou vyhledávány jednotlivá slova i celé fráze, vzdálenosti mezi nimi a váhy shody. Dotazy jsou velmi rychle provedené, vrací méně a přesnějších výsledků, ale mají

problém s hledáním „nevýznamných slov“. Např. když je třeba najít frázi „dobře, ale dotaz“, vrátí CONTAINS chybové hlášení, že dotaz obsahuje „nevýznamné slova“, a protože tyto slova nejsou standardně zahrnuta do vyhledávacího procesu, není možné tuto frázi najít, i kdyby skutečně v některém textu byla.

CONTAINS může vyhledávat:

- Slovo nebo fráze
- Předponu slova nebo fráze
- Slovo blízko jiného slova (vzdálenost).
- Slovo, které má vyšší váhu než jiné slovo

## Syntaxe

```
CONTAINS
  ( { sloupec | * } , '< podmínka >'
    )

< podmínka > ::=
  { < jednoduchý výraz >
    | < výraz předložky >
    | < výraz podobnosti >
    | < výraz váhy >
    }
  | { ( < podmínka > )
    { AND | AND NOT | OR } < podmínka > [ ...n ]
    }

< jednoduchý výraz > ::=
  slovo | " fráze "

< výraz předložky > ::=
  { " slovo * " | " fráze * " }

< výraz podobnosti > ::=
  { < jednoduchý výraz > | < výraz předložky > }
  { { NEAR | ~ } { < jednoduchý výraz > | < výraz předložky > } } [ ...n
  ]

< výraz váhy > ::=
  ISABOUT
  ( { {
    < jednoduchý výraz >
    | < výraz předložky >
    | < výraz podobnosti >
    }
    [ WEIGHT ( váha ) ]
    } [ ,...n ]
  )
```

## Parametry

*sloupec*

Je jméno sloupce určeného pro fulltextové vyhledávání. Všechny řetězcové sloupce jsou vhodné pro fulltextové vyhledávání

\*

Určuje, že všechny sloupce v tabulce určené pro fulltextové vyhledávání mohou být použity při hledání shody s danými vyhledávacími podmínkami.

<podmínka >

Určíte text hledaný v tabulce.

*Slovo*

Je řetězec bez uvozovek nebo apostrofů.

*fráze*

Jedno nebo více slov s mezerami mezi sebou.

### **Použití CONTAINS s jednoduchým výrazem**

Tento příklad najde všechny produkty s cenou \$15.00, které obsahují slovo "bottles".

```
USE Northwind
GO
SELECT ProductName
FROM Products
WHERE UnitPrice = 15.00
      AND CONTAINS(QuantityPerUnit, 'bottles')
GO
```

### **Použití CONTAINS a fráze v jednoduchém výrazu**

Tento příklad najde všechny produkty, které obsahují frázi "sasquatch ale" nebo "steeleye stout".

```
USE Northwind
GO
SELECT ProductName
FROM Products
WHERE CONTAINS(ProductName, ' "sasquatch ale" OR "steeleye stout" ')
GO
```

### **Použití CONTAINS s předložkovým výrazem**

Tento příklad najde všechna jména produktů s aspoň jedním slovem začínajícím předložkou „choc“ ve sloupci **ProductName**.

```
USE Northwind
GO
SELECT ProductName
FROM Products
WHERE CONTAINS(ProductName, ' "choc*" ')
GO
```

### **Použití CONTAINS a OR s předložkovým výrazem**

Tento příklad najde všechny popisy kategorií, které obsahují řetězec "sea" or "bread".

```
USE Northwind
SELECT CategoryName
```

```
FROM Categories
WHERE CONTAINS(Description, '"sea*' OR "bread*"' )
GO
```

### Použití CONTAINS s výrazem podobnosti

Tento příklad najde všechny názvy produktů, kde je slovo "Boysenberry" blízko slova "spread".

```
USE Northwind
GO
SELECT ProductName
FROM Products
WHERE CONTAINS(ProductName, 'spread NEAR Boysenberry')
GO
```

### Použití CONTAINS s váhovým výrazem

Tento příklad najde všechny názvy produktů obsahujících slova spread, sauces, nebo relishes a každé slovo má jinou váhu.

```
USE Northwind
GO
SELECT CategoryName, Description
FROM Categories
WHERE CONTAINS(Description, 'ISABOUT (spread weight (.8),
    sauces weight (.4), relishes weight (.2) )' )
GO
```

### Použití CONTAINS s proměnnými

Tento příklad využívá proměnných namísto určitého vyhledávacího výrazu.

```
USE pubs
GO
DECLARE @SearchWord varchar(30)
SET @SearchWord = 'Moon'
SELECT pr_info FROM pub_info WHERE CONTAINS(pr_info, @SearchWord)
```

## 2.2. CONTAINSTABLE

Vrací tabulku výsledků naplněnou řádky takových sloupců, které obsahují texty s naprostou shodou, nebo fuzzy (méně přesnou) shodou, jsou vyhledávány jednotlivá slova i celé fráze, vzdálenosti mezi nimi a váhy shody. CONTAINSTABLE může být uvedena v části WHERE příkazu SELECT.

Dotazy používající CONTAINSTABLE vrací rank pro každý řádek.

### Syntaxe

```
CONTAINSTABLE ( tabulka , { sloupec | * } , ' < podmínka > '
    [ , prvních n ] )
```

```
< podmínka > ::=
    { < jednoduchý výraz >
    | < předložkový výraz >
    | < výraz podobnosti >
```

```

    | < váhový výraz >
    }
    | { ( < podmínka > )
    { AND | AND NOT | OR } < podmínka > [ ...n ]
    }

< jednoduchý výraz > ::=
    slovo | " fráze "

< předložkový výraz > ::=
    { "slovo * " | "fráze * " }

< výraz podobnosti > ::=
    { < jednoduchý výraz > | < předložkový výraz > }
    { { NEAR | ~ } { < jednoduchý výraz > | < předložkový výraz > } } [
...n ]

< váhový výraz > ::=
    ISABOUT
    ( { {
        < simple_term >
        | < předložkový výraz >
        | < výraz podobnosti >
        }
    [ WEIGHT ( váha ) ]
    } [ ,...n ]
    )

```

## Parametry

*tabulka*

Jméno tabulky, která je připravena na fulltextové vyhledávání.

*prvních n*

Určuje, že pouze prvních n nejvýše oznámkovaných výsledku v klesajícím pořadí budou dotazem vráceny.

```

SELECT select_list
FROM table AS FT_TBL INNER JOIN
    CONTAINSTABLE(table, column, contains_search_condition) AS KEY_TBL
ON FT_TBL.unique_key_column = KEY_TBL.[KEY]

```

## Příklady využití CONTAINSTABLE

Tento příklad hledá všechna jména produktů obsahující slova breads, fish, nebo bedra a každé slovo má jinou váhu. U každého řádku ve výsledku je uveden rank. Řádky s vyšším rankem jsou uvedeny přednostně

```

USE Northwind
GO
SELECT FT_TBL.CategoryName, FT_TBL.Description, KEY_TBL.RANK
FROM Categories AS FT_TBL INNER JOIN
    CONTAINSTABLE(Categories, Description,
        'ISABOUT (bread weight (.8),
        fish weight (.4), beers weight (.2) )' ) AS KEY_TBL
ON FT_TBL.CategoryID = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
GO
-- nebo
SELECT [key], [rank]

```



```
FROM CONTAINSTABLE(Articles, *, 'FORMSOF(INFLECTIONAL('Ichiro')
AND "sports"')
```

### Dotaz s uvedením podmínky „větší než“

Tento příklad vrací popis a jméno kategorie všech food kategorií, kde popis obsahuje slova "sweet and savory" blízko slova "sauces" nebo "candies". Všechny řádky s kategorií "Seafood" jsou zahazeny a vráceny jsou pouze řádky s rankem větším než 2.

```
USE Northwind
GO
SELECT FT_TBL.Description,
       FT_TBL.CategoryName,
       KEY_TBL.RANK
FROM Categories AS FT_TBL INNER JOIN
     CONTAINSTABLE (Categories, Description,
                    '("sweet and savory" NEAR sauces) OR
                    ("sweet and savory" NEAR candies) '
                    ) AS KEY_TBL
     ON FT_TBL.CategoryID = KEY_TBL.[KEY]
WHERE KEY_TBL.RANK > 2
     AND FT_TBL.CategoryName <> 'Seafood'
ORDER BY KEY_TBL.RANK DESC
```

### Dotaz s uvedením pouze prvních deseti výsledků

Tento příklad vrací popis a jméno kategorie prvních deseti food kategorií, kde popis obsahuje slova "sweet and savory" blízko slova "sauces" nebo "candies".

```
SELECT FT_TBL.Description,
       FT_TBL.CategoryName,
       KEY_TBL.RANK
FROM Categories AS FT_TBL INNER JOIN
     CONTAINSTABLE (Categories, Description,
                    '("sweet and savory" NEAR sauces) OR
                    ("sweet and savory" NEAR candies) '
                    , 10
                    ) AS KEY_TBL
     ON FT_TBL.CategoryID = KEY_TBL.[KEY]
```

## 2.3. FREETEXT

Je používán ke hledání sloupců obsahujících řetězce, které obsahují texty, které významově odpovídají zadaným kritériím a nehledá striktně pouze slova uvedená v kritériích. FREETEXT při spuštění láme slova na jednotlivé výrazy a přiřazuje jim váhu a na konci vrátí výsledky.

### Syntaxe

```
FREETEXT ( { sloupec | * } , 'freetext řetězec' )
```

### Parametry

*freetext řetězec*

Text hledaný v daném sloupci. Může to být jakýkoliv text jako slova, fráze, věty.

### Použití FREETEXTu na hledání slov

Tento příklad hledá všechny kategorie produktů, které obsahují slova významově podobná slovům bread, candy, dry, a meat v popisu produktu jako je breads, candies, dried, a meats.

```
USE Northwind
GO
SELECT CategoryName
FROM Categories
WHERE FREETEXT (Description, 'sweetest candy bread and dry meat' )
GO
```

## 2.4. FREETEXTTABLE

Vrací tabulku naplněnou řádky s id takových sloupců v databázi, které obsahují významově podobná slova, které jsou zadané v kritériích a nehledá pouze slova uvedená v těchto kritériích.

Dotazy vrací rank pro každý řádek.

### Syntaxe

```
FREETEXTTABLE ( tabulka , { sloupec | * } , 'freetext řetězec' [ , prvních n ] )
```

### Příklad

Tento příkaz vrací jméno kategorie a popis všech kategoriích, které odpovídají slovům sweet, candy, bread, dry, a meat.

```
USE Northwind
SELECT FT_TBL.CategoryName,
       FT_TBL.Description,
       KEY_TBL.RANK
FROM Categories AS FT_TBL INNER JOIN
     FREETEXTTABLE(Categories, Description,
     'sweetest candy bread and dry meat') AS KEY_TBL
ON FT_TBL.CategoryID = KEY_TBL.[KEY]
GO

-- nebo

SELECT [key], [rank]
FROM FREETEXTTABLE(Articles, Body_Sports, 'Ichiro')
```

## 3. Fulltextové a SQL indexy

Podpora fulltextu v SQL Serveru přináší dvě výhody

- schopnost vypořádat se s dotazy na textová data
- vytvoření a údržba indexů umožňujících zjištění odpovědi na tyto dotazy

Fulltextové indexy se liší od klasických SQL indexů v mnoha ohledech

**Klasické SQL indexy**

**Fulltextové indexy**

Uloženy v kontrolovány databázi v nichž byly vytvořeny	Uloženy v souborech, ale administrovány přes databázi
Je možné vytvořit více indexů v jedné tabulce	Je povolen pouze jeden fulltextový index na tabulku
Automaticky aktualizovány ve chvíli, když jsou data vkládána, měněna nebo mazána	Přidání dat do fulltext indexu může probíhat periodicky po určité době, vyžádáno speciálním požadavkem, nebo nastaveno na automatickou aktualizaci při každé změně
Neseskupené	Seskupené v jedné databázi v jednom nebo více katalozích
Vytvořeny a rušeny v SQL Server Enterprise Manageru, wizardech, nebo Transact-SQL dotazech.	Vytvořeny a rušeny v SQL Server Enterprise Manager, wizardech, nebo uložených procedurách

## 4. Aktualizace indexů

MS SQL Server dokáže aktualizovat data bez jakéhokoliv zásahu pomocí Fulltextového Indexing Wizarda, SQL Server Enterprise Managera nebo SQL Server Agent job schedulera.

Jsou tři možnosti jak nastavit aktualizaci indexů a to

- Plná obnova – znovu-indexování všech záznamů v databázi
- Periodická indexace – po určitých časových intervalech jsou projity záznamy o změnách a k nim příslušné záznamy v tabulce jsou aktualizovány
- Stopování změn – SQL server poskytne seznam všech změn u indexovaných dat a uživatel může sám vybrat, které změny uložit a které ne.

## 5. Konfigurace fulltextového vyhledávání

Na vytvoření fulltextového indexu na textová data uložená v databázi je potřeba projít několik kroků a to

- Jako sysadmin nebo db\_owner pro jednotlivou databázi vykonajte tyto příkazy

```
use Northwind
exec sp_fulltext_database 'enable'
```

- Je třeba vytvořit fulltext katalog k uložení fulltextových indexů. Jak již bylo řečeno, data v tomto katalogu jsou v souborovém systému mimo SQL Server, proto je třeba být opatrný kam tyto data uložit. Pokud nebude specifikováno jinak, bude katalog uložen do podadresáře FTDAT v Microsoft SQL Server\MSSQL adresáři. Zde je příklad jak uložit data na jiné umístění

```
exec sp_fulltext_catalog 'Cat_Desc', 'create', 'f:\ft'
```

Konvence pojmenování katalogů v MSS je

```
SQL+dbid+catalogID
```

- Pokud to tak již není, vytvořte unikátní nenulový sloupec v každé tabulce, ve které chcete vytvořit fulltextový index. Tento unikátní index je použit k mapování jednotlivých řádků na unikátní stlačitelný klíč používaný pomocí MSS.
- Dále je třeba dát vědět MSS, že chcete vytvořit fulltextový index na tabulky. Následujícím příkazem bude tabulka přidána do fulltext katalogu (v tomto případě, 'Cat\_Desc', kterou jsme předtím vytvořili):

```
exec sp_fulltext_table 'Categories', 'create', 'Cat_Desc',  
'PK_Categories'
```

- Dalším krokem je přidání sloupců do fulltextového indexu. Pro každý sloupec je možné zvolit jazyk a jestliže je sloupec typu image, je třeba specifikovat další sloupec, který bude použit pro zjištění typu dokumentu každého řádku.

```
exec sp_fulltext_column 'Categories', 'Description', 'add'
```

Pokud není zvolen jazyk, ve kterém jsou informace ve sloupci napsané, je zvolen implicitní jazyk systému. Pokud je třeba ho změnit, použije se příkaz "sp\_configure".

- Nyní je již fulltextový index připraven. Pro naplnění daty použijeme následující příkaz, který vynutí naplnění indexu.

```
exec sp_fulltext_table 'Categories', 'start_full'
```

- Pokud Vás zajímá stav indexování tabulky, použijte příkaz

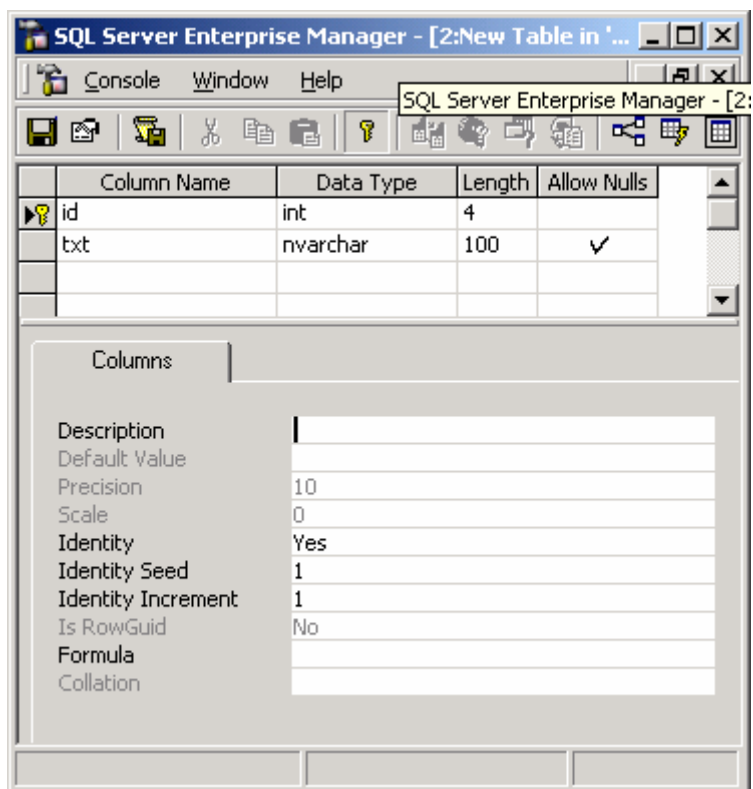
```
select FULLTEXTCATALOGPROPERTY('Cat_Desc', 'Populatestatus')
```

## 5.1. Příklad fulltextového vyhledávání v MS SQL Serveru

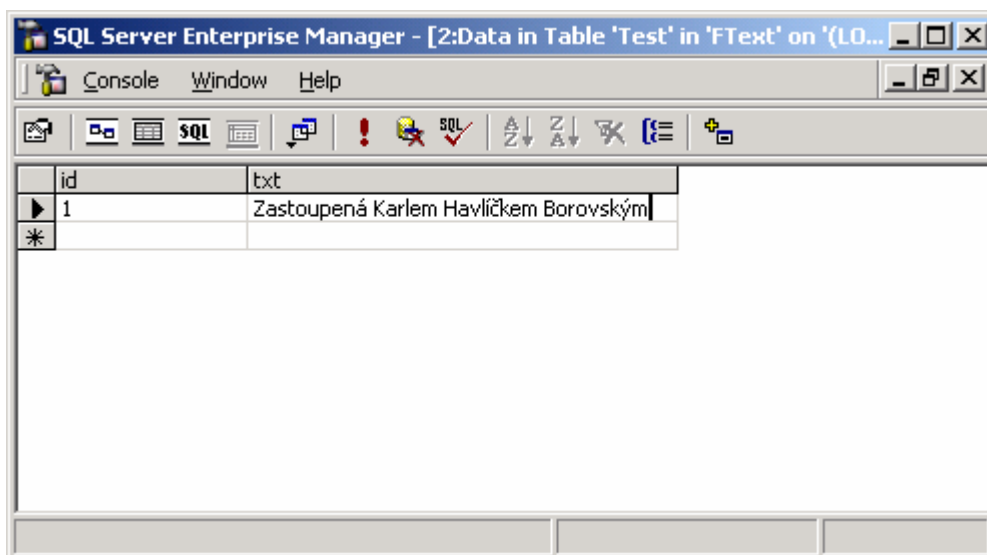
Nejprve vytvoříme novou tabulku se sloupci:

- *id* – typ *integer*, primární klíč, *Identity = Yes*
- *txt* – typ *nvarchar(100)*

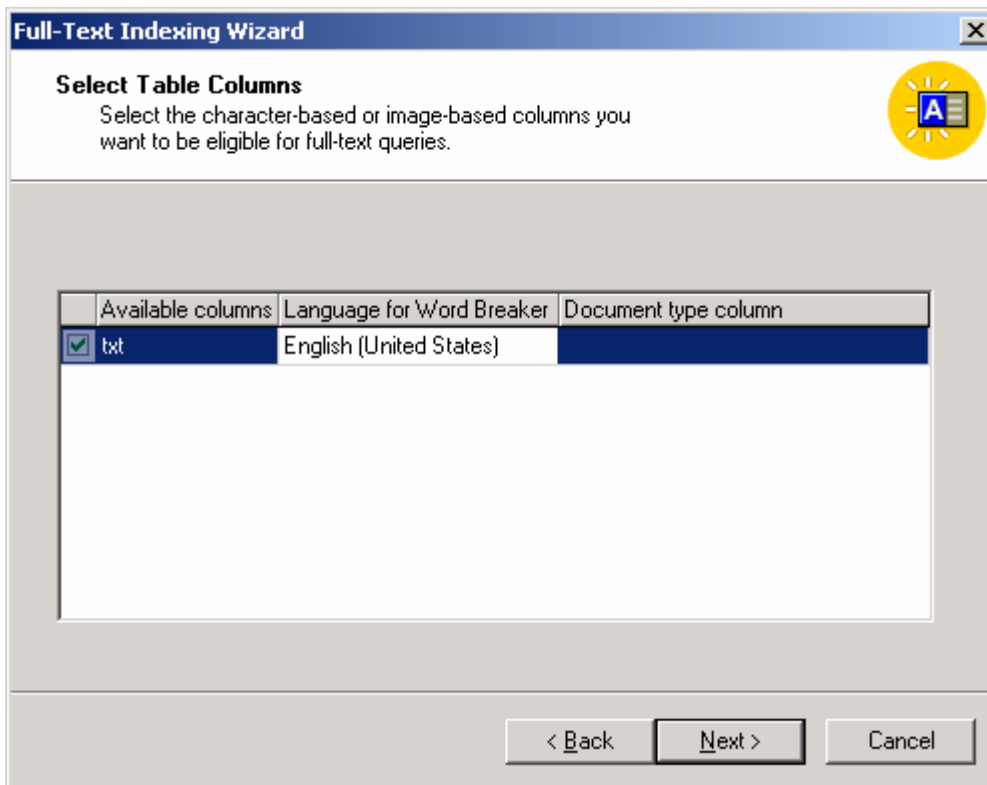
Při ukládání tabulku pojmenujte *Test*:



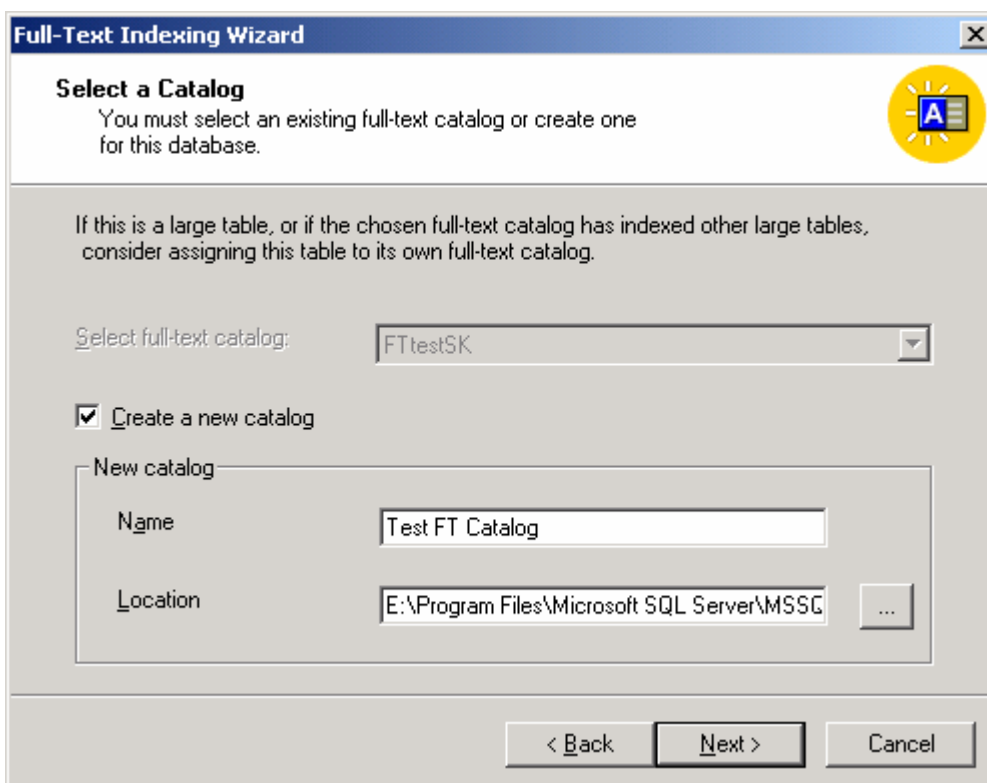
Nyní zadáme testovací údaj. Klikněte pravým tlačítkem na tabulce, zvolte *Open Table, Return all rows* a zadejte nový řádek se sloupcem *txt* rovným např. „zastoupená Karlem Havlíčkem Borovským“:



Full-textové vyhledávání definujeme kliknutím pravým tlačítkem na tabulce a volbou *Full Text Index Table, Define Full-Text Indexing on a Table*. Jako unikátní klíč v tabulce by se měl nabídnout primární klíč tabulky *PK\_Test*. Na další stránce zvolte indexaci sloupce *txt* a jazyk prozatím nastavte na *English (United States)*. Důvodem je, že seznam jazyků je v této verzi nastaven „natvrdo“, při používání jiného jazyka je nutné použít systémové uložené procedury pro změnu používaného jazyka:

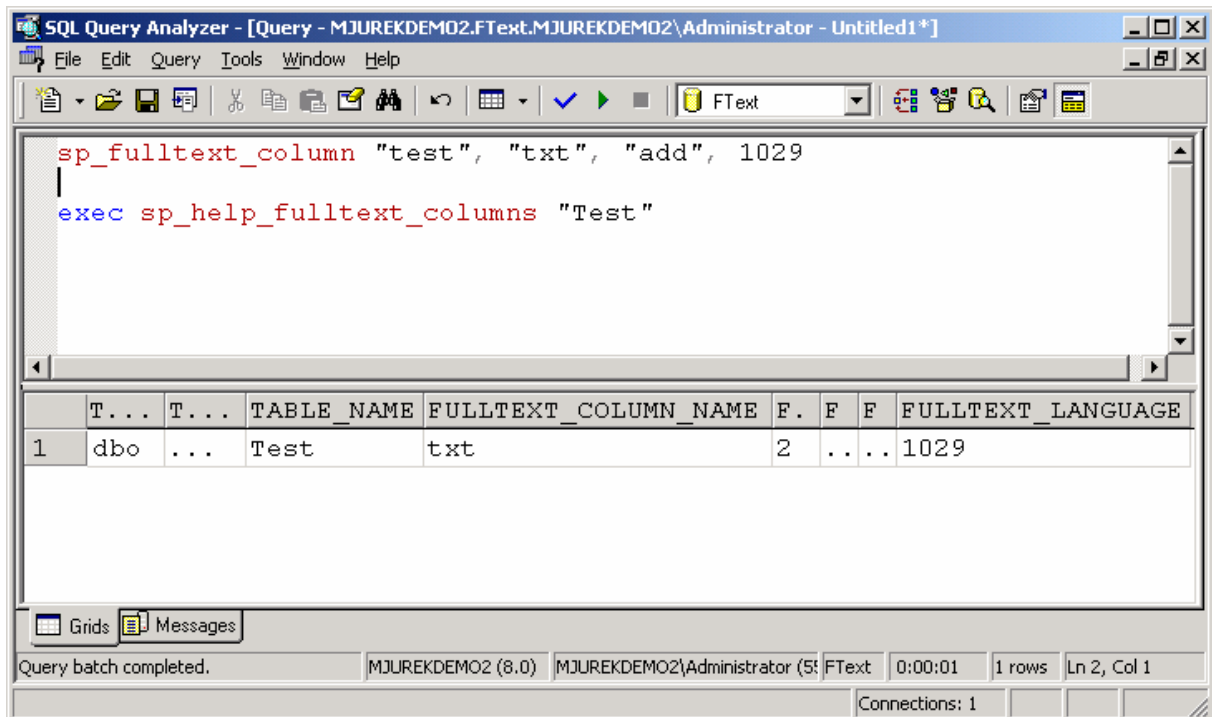


Na další stránce vyberte vytvoření nového katalogu a nazvěte jej například *Test FT Catalog*:

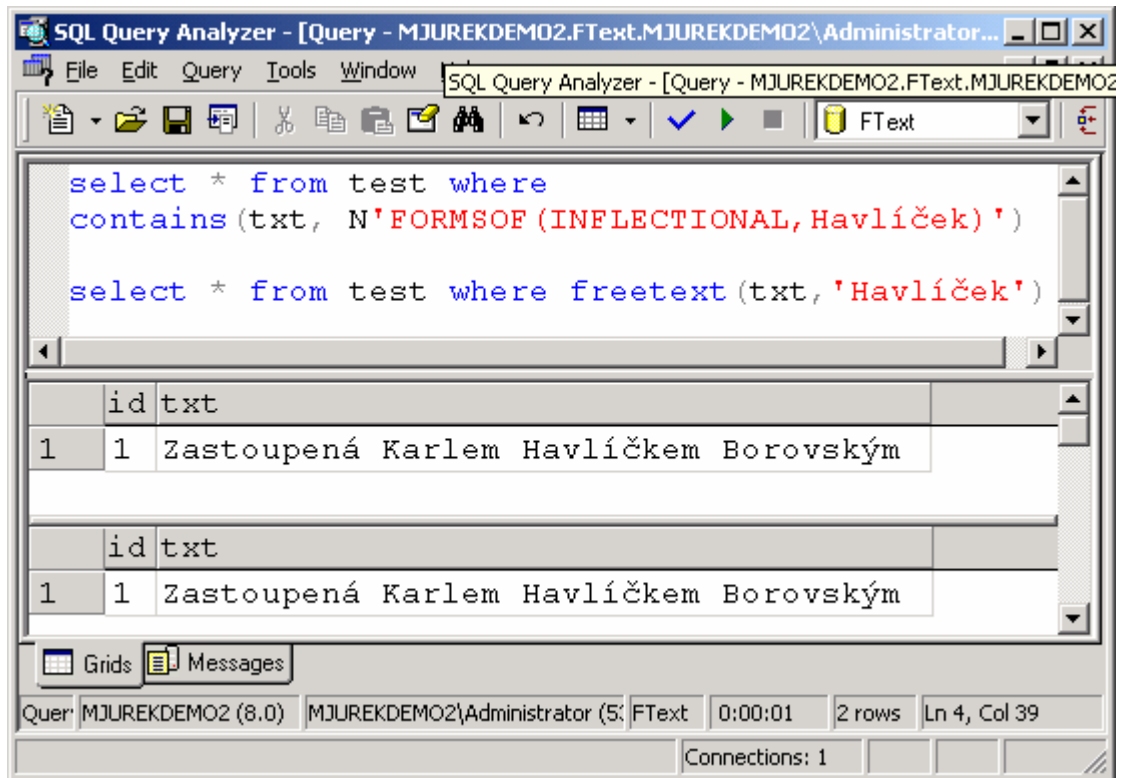


Zbytek průvodce dokončete beze změn výchozích hodnot. Nyní je třeba přenastavit jazyk indexovaných sloupců. Z menu *Tools, Query Analyzer* spusťte nástroj *SQL Query Analyzer*, zkontrolujte, zda spojení používá správnou databázi. K nastavení správného jazyka pro indexovaný sloupec lze použít uloženou proceduru *sp\_fulltext\_column*, pro zjištění právě

nastaveného jazyka lze použít uloženou proceduru *sp\_help\_fulltext\_columns*. Jako jazyk se používá číselně vyjádřený kód místního nastavení, což je 1029 pro češtinu a 1051 pro slovenštinu. Následující příkaz nastaví jazyk sloupce *txt* na češtinu a provede kontrolu nastavení:



Změna jazyka se v katalogu neprojeví automaticky, proto je nutné katalog znovu vytvořit. Běžte v *Enterprise Manageru* do složky *Full-Text Catalogs*, klikněte pravým tlačítkem na katalogu *Test FT Catalog* a zvolte *Start Full Population*. Počkejte několik vteřin – stav procesu můžete kontrolovat v *Enterprise Manageru*, ale okno je nutné obnovovat ručně. Při správné funkci fulltextového vyhledávání vrátí libovolný z následujících dotazů řádek obsahující slovo „Havlíček“ v libovolném tvaru:



## 6. Odkazy

- MSDN
- Full-Text Search Public Newsgroup (microsoft.public.sqlserver.fulltext)
- SQL Server development team