Demo 1 - Creating the Catalog

1. SQL Server Express with Advanced Services SP2 http://msdn2.microsoft.com/en-us/express/bb410792.aspx
2. Open SQL Server Management Studio.
3. Open New Query
4. Make sure AdventureWorks is the selected database.
5. Create the catalog:

```
create fulltext catalog AdvWorksCatalog as default
```

Demo 2 - Creating the Index

1. We'll be using the Production.ProductDescription table.
2. Show the Keys area to show Primary Key

```
create fulltext index on Production.ProductDescription
  ([Description])
  key index PK_ProductDescription_ProductDescriptionID
  on AdvWorksCatalog
  with change_tracking auto
```

Demo 3 - Query the Index

1. Start with CONTAINS.

```
select [Name], ProductNumber, [Description]
from [Production].[Product] p
   , [Production].[ProductDescription] pd
   , [Production].[ProductModelProductDescriptionCulture] pmpdc
where p.ProductModelID = pmpdc.ProductModelID
  and pmpdc.ProductDescriptionID = pd.ProductDescriptionID
  and CONTAINS(pd.[Description], 'shifting')
```

2. Demonstrate the exact match feature by shifting vs shift

Demo 4 - FreeText

1. Now show FREETEXT

```
select [Name], ProductNumber, [Description]
from [Production].[Product] p
   , [Production].[ProductDescription] pd
   , [Production].[ProductModelProductDescriptionCulture] pmpdc
where p.ProductModelID = pmpdc.ProductModelID
  and pmpdc.ProductDescriptionID = pd.ProductDescriptionID
  and FREETEXT(pd.[Description], 'shift')
```

Demo 5 – Advanced Querying

1. Show FORMSOF INFLECTIONAL

```
select [Name], ProductNumber, [Description]
from [Production].[Product] p
   , [Production].[ProductDescription] pd
   , [Production].[ProductModelProductDescriptionCulture] pmpdc
where p.ProductModelID = pmpdc.ProductModelID
  and pmpdc.ProductDescriptionID = pd.ProductDescriptionID
  and CONTAINS(pd.[Description], 'FORMSOF(INFLECTIONAL, light)' )
```

2. Talk about FORMSOF THESSAURUS, no good data for example

3. Phrases – wrap in double quotes

```
select [Name], ProductNumber, [Description]
from [Production].[Product] p
   , [Production].[ProductDescription] pd
   , [Production].[ProductModelProductDescriptionCulture] pmpdc
where p.ProductModelID = pmpdc.ProductModelID
  and pmpdc.ProductDescriptionID = pd.ProductDescriptionID
  and CONTAINS(pd.[Description], '"stiff ride"' )
```

4. NEAR

```
select [Name], ProductNumber, [Description]
from [Production].[Product] p
   , [Production].[ProductDescription] pd
   , [Production].[ProductModelProductDescriptionCulture] pmpdc
where p.ProductModelID = pmpdc.ProductModelID
  and pmpdc.ProductDescriptionID = pd.ProductDescriptionID
  and CONTAINS(pd.[Description], 'safe near ride' )
```

5. OR

```
select [Name], ProductNumber, [Description]
from [Production].[Product] p
   , [Production].[ProductDescription] pd
   , [Production].[ProductModelProductDescriptionCulture] pmpdc
where p.ProductModelID = pmpdc.ProductModelID
  and pmpdc.ProductDescriptionID = pd.ProductDescriptionID
  and CONTAINS(pd.[Description], 'safe or ride' )
```

6. Prefixed Terms (term plus wildcard - *)

```
select [Name], ProductNumber, [Description]
from [Production].[Product] p
   , [Production].[ProductDescription] pd
   , [Production].[ProductModelProductDescriptionCulture] pmpdc
where p.ProductModelID = pmpdc.ProductModelID
  and pmpdc.ProductDescriptionID = pd.ProductDescriptionID
  and CONTAINS(pd.[Description], '"light*"' )
```

Demo 6 - Query with Ranking

```
-- Simple Example
select [KEY], RANK
from freetexttable([Production].[ProductDescription], [Description], 'light' )

-- rank: higher number is less precise - low is good
-- key/rank is case sensitive
select fts.[KEY], fts.[RANK], [Description]
from [Production].[ProductDescription] as pd
inner join freetexttable([Production].[ProductDescription], [Description], 'light' )
as fts on fts.[KEY] = pd.ProductDescriptionID
order by fts.[RANK]

-- Complex example
select fts.[KEY], fts.[RANK], [Name], ProductNumber, [Description]
from [Production].[ProductDescription] pd
inner join freetexttable([Production].[ProductDescription], [Description], 'light' )
as fts on fts.[KEY] = pd.ProductDescriptionID
inner join [Production].[ProductModelProductDescriptionCulture] as pmpdc on
pmpdc.ProductDescriptionID = pd.ProductDescriptionID
inner join [Production].[Product] as p on p.ProductModelID = pmpdc.ProductModelID
```

Demo 7 – Use inside and Application

Show WPF_Binding_To_Full_Text_Search.sln, use keyword of **feature** to demonstrate difference between exact match.

Demo 8 – Helpful System Queries

```
-- List names of all catalogs attached to the database
select * from sys.fulltext_catalogs

-- List names of all FTS indexes
select cat.[name] as CatalogName
     , object_name(object_id) as table_name
     , is_enabled
     , change_tracking_state_desc
  from sys.fulltext_indexes, sys.fulltext_catalogs cat

-- List table name and column number (ordinal position) for indexes
select object_name(object_id) table_name, column_id from sys.fulltext_index_columns
```

Demo 9 – Clean up

drop fulltext index on table_name
drop fulltext catalog catalog_name

```
drop fulltext index on [Production].[ProductDescription];
drop fulltext catalog AdvWorksCatalog;
```

Note that it is not required to remove all indexes before you remove a catalog, removing the catalog automatically kills all associated indexes.