Check if you have the thing:

```
SELECT FULLTEXTSERVICEPROPERTY('IsFullTextInstalled');
```

Make a full text Catalog:

1. Open ssms
2. [Desired Database] -> Right-Click Storage -> New Full-Text Catalog (Full-Text Catalog opens)

2(Alt) Create it in code

```
CREATE FULLTEXT CATALOG [putnamehere]
WITH ACCENT_SENSITIVITY = OFF
AS DEFAULT
AUTHORIZATION dbo
```

Make Full-Text Index

1. Select desired table
2. Right-Click -> Full-Text Index -> Define Full-Text Index
3. Choose primary key
4. Choose columns to be searched
5. Select change tracking to maintain a changelog to update full index. (Automatic)
6. Select your catalog
7. Select full text index population schedule

Full-Text with SQL

```
CREATE FULLTEXT INDEX
ON Production.ProductModel
(
  CatalogDescription LANGUAGE English,
  Instructions LANGUAGE English,
  Name LANGUAGE English
  )

KEY INDEX PK_ProductModel_ProductModelID
ON
(
  AdventureWorksFTCat
)
WITH
(
CHANGE_TRACKING AUTO
);
GO

ALTER FULLTEXT INDEX
ON Production.ProductModel ENABLE;
GO
```

FREETEXT : searches for words that match inflectional forms and thesaurus expansions and replacements.

Example

```
SELECT
   ProductModelID,
   Name,
   CatalogDescription,
   Instructions
FROM Production.ProductModel
WHERE FREETEXT(*, N'sock');
```

! You can add an optional language parameter at the end for language. English is “LANGUAGE 1033”

CONTAINS: More advanced form of Freetext

Example Simple:

```

SELECT
  ProductModelID ,
  Name,
  CatalogDescription,
  Instructions
FROM Production.ProductModel
WHERE CONTAINS (*, N'weld');
```

Example Look in columns for any form of a word

```
SELECT
  ProductModelID ,
  Name,
  CatalogDescription
FROM Production.ProductModel
WHERE CONTAINS
(
  (
    Name,
    CatalogDescription
  ),
  N'FORMSOF(INFLECTIONAL, sport)'
);
```

Must contain all (switch | with &)

```
SELECT
  ProductModelID ,
  Name,
  CatalogDescription
FROM Production.ProductModel
WHERE CONTAINS
(
  (
    Name,
    CatalogDescription
),
N'"tube" | FORMSOF (INFLECTIONAL, sport)'
);
```

!You can also replace “INFLECTIONAL” with “THESAURUS” for synonyms
! Use &! for AND NOT

Find everything that begins with “bot”

```
SELECT
  ProductModelID ,
  Name
FROM Production.ProductModel
WHERE CONTAINS (Name, N'"bot*"');
```

Search instructions where the word aluminum is within 3 words of jig

```

SELECT
  ProductModelID ,
  Name
FROM Production.ProductModel
WHERE CONTAINS(Instructions, 'NEAR((aluminum,jig), 3)');
```

!Add TRUE as the third parameter to enforce that aluminum comes before jig.

FREETEXTTABLE and CONTAINSTABLE work like the previous two predicates, but returns tables with a KEY and RANK. KEY is the primary key of the table, rank is the “closeness to the match”

Looks for aluminum in instructions in ProductModel

```
SELECT
  ftt.[KEY],
  ftt.[RANK],
  pm.ProductModelID ,
  pm.Name FROM FREETEXTTABLE
(
  Production.ProductModel,
  Instructions,
  N'aluminum'
) ftt
INNER JOIN Production.ProductModel pm
  ON ftt.[KEY] = pm.ProductModelID;
```

Returns a table of searches that find tours in [Name] in ProductModel

```
SELECT
  [KEY],
  [RANK]
FROM CONTAINSTABLE (
Production.ProductModel,
[Name],
N'FORMSOF(INFLECTIONAL, tours)'
);
```

Set up a table for Thesaurus Use

```
CREATE FULLTEXT INDEX ON Production.Product
(
    Name  LANGUAGE  English,
    Color LANGUAGE English
)
KEY  INDEX PK_Product_ProductID
ON  (AdventureWorksFTCat)
WITH
(
    CHANGE_TRACKING AUTO,
    STOPLIST = SYSTEM
);
GO

ALTER FULLTEXT INDEX ON Production.Product
ENABLE;
GO
```

Reload your thesaurus (if you’ve tampered with it)

```
EXEC sys.sp_fulltext_load_thesaurus_file 1033;
GO
```

Come back to:

ISABOUT

Lets you apply a “weight” between 0.0 and 1.0 in CONTAINSTABLE to whatever you’re looking up. It said something about applying it to the RANK but I’m not 100% sure what it does exactly.

Messing with the Thesaurus

Stop Words and Stop Lists

Stored procedures that let you see how freetext is malfunctioning

“Statistical Semantics”