

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE TECNOLOGIA**

**CONSULTAS POR CONTEÚDO A TEXTOS POR MEIO DA  
SIMILARIDADE DE CONCEITOS USANDO RECURSOS DO SISTEMA  
DE GERENCIAMENTO DE BANCOS DE DADOS SQL SERVER**

BRENDA CRISTINA DE SOUZA SILVA

LIMEIRA, 2021

BRENDA CRISTINA DE SOUZA SILVA

**CONSULTAS POR CONTEÚDO A TEXTOS POR MEIO DA  
SIMILARIDADE DE CONCEITOS USANDO RECURSOS DO SISTEMA  
DE GERENCIAMENTO DE BANCOS DE DADOS SQL SERVER**

Monografia apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como requisito para obtenção do título de Bacharel em Sistemas de Informação.

**Orientadora: Profa. Dra. Gisele Busichia Baioco**

LIMEIRA

2021

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Faculdade de Tecnologia  
Luiz Felipe Galeffi - CRB 8/10385

Si38c Silva, Brenda Cristina de Souza, 1999-  
Consultas por conteúdo a textos por meio da similaridade de conceitos usando recursos do sistema de gerenciamento de bancos de dados SQL Server / Brenda Cristina de Souza Silva. – Limeira, SP : [s.n.], 2022.

Orientador: Gisele Busichia Baioco.  
Trabalho de Conclusão de Curso (graduação) – Universidade Estadual de Campinas, Faculdade de Tecnologia.

1. Recuperação da informação. 2. SQL server. 3. Pesquisa por palavra-chave.  
I. Baioco, Gisele Busichia, 1970-. II. Universidade Estadual de Campinas. Faculdade de Tecnologia. III. Título.

Informações adicionais, complementares

**Título em outro idioma:** Content based text queries by concept similarity using SQL Server database management system features

**Palavras-chave em inglês:**

Information retrieval

SQL server

Keyword searching

**Titulação:** Bacharel

**Banca examinadora:**

Gisele Busichia Baioco [Orientador]

Antonio Carlos Zambon

Luiz Camolesi Júnior

**Data de entrega do trabalho definitivo:** 29-06-2022

## **AGRADECIMENTOS**

Agradeço primeiramente a minha família, que sempre me apoiou e incentivou independente de tudo. Aos meus amigos que participaram dessa jornada cheia de obstáculos e desafios.

Agradeço especialmente à minha orientadora que me auxiliou a solucionar as dificuldades encontradas no caminho.

Agradeço a mim, porque eu não desisti.

*“Você nunca sabe que resultados  
virão da sua ação.  
Mas se você não fizer nada,  
não existirão resultados.”*

Mahatma Gandhi

## RESUMO

O conceito de Recuperação de Informação é o processo de recuperar documentos de uma coleção em resposta a uma consulta. Esse processo é capaz de localizar tipos específicos de documentos como, por exemplo, manuscritos, livros, páginas web, e-mails, mensagens instantâneas, entre outros. Com o objetivo de recuperar documentos por meio da similaridade de conceitos este trabalho apresenta o estudo de dados do tipo *Full Text*, utilizando recursos oferecidos pelo Sistema de Gerenciamento de Banco de Dados SQL Server. Os recursos foram aplicados em um estudo de caso que utiliza busca semântica e pesquisas a texto completo. A partir das análises dos resultados foram propostas consultas adequadas para realizar busca de documentos utilizando similaridade de conceitos.

**Palavras-chave:** Recuperação de Informação, SQL Server, *Full Text*, busca semântica.

## **ABSTRACT**

The concept of Information Retrieval is the process of retrieving documents from a collection in response to a query. This process can find specific types of documents, such as manuscripts, books, web pages, emails, instant messages, among others. With the objective of retrieving documents through the similarity of concepts, this work presents the study of Full Text data, using resources offered by the SQL Server Database Management System. The resources were applied in a case study that uses semantic search and full-text searches. Based on analysis of the results, appropriate queries were proposed to search for documents using similarity of concepts.

**Keywords:** Retrieval Information, SQL Server, Full Text, Semantic Search.

## SUMÁRIO

1. INTRODUÇÃO.....	9
2. FUNDAMENTAÇÃO TEÓRICA.....	11
2.1. Recuperação de Informação.....	11
2.2. SQL/MM .....	13
3. RECURSOS DO SQL SERVER PARA DADOS <i>FULL TEXT</i> .....	18
4. ESTUDO DE CASO.....	25
4.1 Proposta de solução .....	36
4.2 Resultados obtidos .....	40
5. CONSIDERAÇÕES FINAIS.....	41
APÊNDICE A .....	46
APÊNDICE B .....	52
APÊNDICE C .....	56

## 1. INTRODUÇÃO

Os bancos de dados multimídia fornecem recursos que permitem aos usuários armazenar e consultar diferentes tipos de dados multimídia, que incluem imagens, vídeos, áudios e textos (textos completos de artigos, livros, mensagens etc). Nesses bancos de dados, as consultas devem ser baseadas no conteúdo dos dados multimídia, ou seja, deve-se utilizar recuperação baseada em conteúdo (*content-based retrieval*), sendo necessário organizar e indexar os dados multimídia com base em seus conteúdos (ELMASRI e NAVATHE, 2016).

Tradicionalmente, os Sistemas de Gerenciamento de Bancos de Dados Relacionais (SGBDR) tratam dados de tipos numéricos ou textuais curtos, estruturados de acordo com o modelo relacional. Para aumentar a flexibilidade dos SGBDR e, conseqüentemente, facilitar seu uso para a manipulação de dados mais complexos, como os dados multimídia, eles têm sido estendidos para incorporar recursos dos modelos orientados a objetos, sendo denominados Sistemas de Gerenciamento de Bancos de Dados Objeto-Relacionais (SGBDOR). Diante desse cenário, a linguagem SQL teve sua terceira versão, denominada SQL:1999 (EISENBERG e MELTON, 1999), desenvolvida principalmente para suportar as extensões objeto-relacionais. Baseado na SQL:1999, surgiu também a SQL *Multimedia and Application Packages*, conhecida como SQL/MM, para tratamento de dados multimídia (MELTON e EISENBERG, 2001).

Dados multimídia do tipo texto, denominados *Full Text*, consistem no texto completo de documentos como artigos, livros, revistas e textos longos em geral. Esse tipo de dado difere das tradicionais cadeias de caracteres em relação ao seu tamanho e por permitir a aplicação de operações específicas, como a busca por palavras específicas, por palavras derivadas (por exemplo, o pretérito de um verbo e o plural de um substantivo), por palavras com significados similares e a busca por palavras que são pronunciadas de maneira semelhante. Outra diferença se refere à indexação. Enquanto cadeias de caracteres são indexadas pelo seu conteúdo completo, tipos especiais de índices são definidos para tipos de dados *Full Text*. Esses índices especiais permitem o registro de informações sobre a similaridade de palavras e frases em um mesmo documento ou em documentos diferentes (MELTON e EISENBERG, 2001).

O trabalho de Paiva (2020) apresenta uma análise dos principais SGBDOR em relação ao suporte a dados do tipo *Full Text* de acordo com as especificações do padrão SQL/MM. Foi realizado um estudo de caso usando o SQL Server que possui o recurso de pesquisa semântica, bem como outros recursos relacionados a tipos de dados *Full Text*, no contexto de avaliação de respostas a questões discursivas.

Diante desse cenário, o objetivo deste trabalho foi aprofundar o estudo dos recursos do SQL Server para suporte a dados *Full Text* e pesquisa semântica de modo a realizar a recuperação de textos por meio da similaridade de conceitos. Usualmente um texto é indexado por palavras-chave e suas frequências relativas. Entretanto, a proposta deste trabalho é a utilização de conceitos, uma vez que um conceito é uma representação escrita de um objeto (tangível) ou de um sentimento (intangível), podendo ser um substantivo ou um sintagma nominal. Com base nos recursos do SQL Server estudados, um estudo de caso foi realizado com o objetivo de desenvolver consultas adequadas para a recuperação de documentos por meio de conceitos.

O trabalho foi organizado com os seguintes capítulos, além da introdução que apresenta as considerações iniciais sobre o tema:

**Capítulo 2 – Fundamentação Teórica:** conceito de Recuperação de Informação e padrões SQL 1999 e SQL/MM.

**Capítulo 3 – Recursos do SQL Server para dados *Full Text*:** apresentação dos recursos presentes no SQL Server para tratamento de dados do tipo *Full Text*.

**Capítulo 4 – Estudo de Caso:** aplicação dos recursos estudados, proposta de consultas para recuperação de textos por meio de conceitos e resultados obtidos.

**Capítulo 5 – Considerações Finais:** apresentação das conclusões que foram obtidas a partir do desenvolvimento do trabalho.

## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1. Recuperação de Informação

O conceito de Recuperação de Informação (RI) é o processo de recuperação de documentos, normalmente de natureza não estruturada, de uma coleção em resposta a uma consulta. Esse processo é capaz de localizar itens específicos de documentos como, por exemplo, manuscritos, documentos, livros, páginas web, e-mails, mensagens instantâneas e coleções de bibliotecas digitais. O campo de RI surgiu a partir da necessidade de realizar consultas a publicações científicas e registros de biblioteca, e se espalhou para outras formas de conteúdo, como arquivos manipulados por médicos, advogados e jornalistas (MANNING; RAGHAVAN; SCHUTZE, 2009).

Dados estruturados apresentam uma estrutura rígida e bem definida, por exemplo registros de uma tabela de um banco de dados relacional, onde os tipos de dados aceitos são previamente determinados durante a criação da tabela. Já os dados semiestruturados, são mais flexíveis, como por exemplo arquivos XML (*Extensible Markup Language*), que possuem *tags* que delimitam os tipos de elementos. E, por fim, os dados não estruturados, que não possuem um modelo bem definido e formal, documentos são exemplos de dados não estruturados (NAVATHE, 2016).

Os sistemas de RI não necessitam que o usuário tenha conhecimento da estrutura ou do conteúdo do banco de dados. A recuperação é realizada a partir das palavras-chaves presentes na solicitação do usuário. Os sistemas de RI são elaborados de acordo com seu objetivo final, como por exemplo, mecanismos de busca dos sistemas operacionais, que buscam por arquivos, pastas e programas.

Segundo Navathe (2016), os sistemas de RI e os bancos de dados estão relacionados dentro do campo da computação. Enquanto os bancos de dados usualmente lidam com dados estruturados, os sistemas de RI lidam com os dados não estruturados, sem um esquema de representação bem definido. Consultas realizadas em bancos de dados retornam uma resposta exata levando em consideração o estado atual do banco e, caso não haja correspondência para uma consulta, o resultado será vazio. Já nos sistemas de RI as consultas são termos (palavras-chave) ou frases da linguagem natural, que resultam em uma lista de identificadores dos documentos, ou trechos de

texto. Para determinar a relevância de cada documento, os sistemas de RI realizam análises estatísticas complexas a partir dos próprios valores dos dados e da frequência de ocorrência dos termos consultados.

No campo de RI há duas abordagens principais, sendo elas a estatística e a semântica. Na primeira são realizadas análises em trechos dos documentos para mensurar sua relevância com a consulta realizada. Já na segunda, abordagem semântica, são utilizadas técnicas de recuperação baseadas em conhecimento fundamentadas em características léxicas, sintáticas do texto e na compreensão do conhecimento, buscam pelo conceito mencionado na consulta, não apenas sua correspondência exata. A fim de aprimorar o processo de RI, a abordagem semântica utiliza também análises estatísticas.

Mesmo com sistemas de RI estatísticos sofisticados, podem não ser recuperados documentos relevantes dado que não capturam o significado ou a informação requerida pelo usuário. Já os modelos de sistemas semânticos são capazes de buscar o significado semântico nos documentos, dessa maneira recuperando documentos que não seriam retornados nos modelos estatísticos. Para isso, são realizadas análises morfológicas, sintáticas e semânticas. Durante as análises morfológicas, são exploradas a raiz, prefixo e sufixo das palavras para avaliar as partes dos documentos. A análise sintática verifica as frases completas e posteriormente a análise semântica que possui diversos métodos que verificam ambiguidades e procuram por sinônimos com base nas relações semânticas entre os trechos dos documentos. Comumente sistemas desse tipo requerem o uso de inteligência artificial e bases de conhecimento complexas.

Para que seja possível realizar o processo de recuperação, primeiramente, os documentos são representados de uma forma adequada. Os termos significativos e suas propriedades são extraídos dos documentos e representados em um índice na forma de uma matriz que contém cada documento em uma linha e as colunas correspondentes armazenam os termos contidos nesse documento. Então, no processo de recuperação de documentos esse índice é utilizado de modo invertido, ou seja, os termos informados em uma consulta são comparados aos termos armazenados na matriz e, quando há

correspondência, o documento é retornado. O resultado será uma lista com os documentos em ordem de relevância (NAVATHE, 2016).

## 2.2. SQL/MM

A linguagem SQL (*Structured Query Language*) foi desenvolvida a partir do modelo relacional proposto por Codd (1970), com o objetivo de definir, modificar e consultar dados em bancos de dados relacionais.

Em 1986 o *American National Standards Institute* (ANSI) publicou o primeiro padrão da linguagem SQL, conhecido como SQL-86, e então, no ano seguinte, a *International Organization for Standardization* (ISO) também reconheceu a SQL como padrão. Desde então diversas atualizações foram lançadas, entre elas, a SQL: 1999 (SQL3), versão que visou atender às necessidades comerciais da época, trazendo importantes recursos, como novos tipos de dados, consultas recursivas, gatilhos e o conceito de orientação a objetos (LIMA, 2004).

Dentre os novos tipos incluídos na versão SQL 1999 estão o tipo *boolean*, *array*, *Large Object* (LOB), *row* e UDT – *User Defined Types*, que são tipos definidos pelo usuário. Este último faz parte dos recursos voltados à padronização do gerenciamento de objetos, cuja estrutura pode conter atributos de qualquer tipo de dado SQL, incluindo outros UDT, além do suporte à herança. Por exemplo, é possível a criação de uma hierarquia, com “pessoa” sendo o supertipo e “estudante” seu subtipo, que terá todos os atributos de pessoa e seus próprios atributos, como “matrícula” e “curso”.

A partir da versão SQL 1999 e os recursos voltados para o conceito de orientação a objetos, foi desenvolvido o padrão SQL/MM (SQL *MultiMedia*), que permite a manipulação de dados multimídia, como imagens, vídeos, dados espaciais e textos completos, chamados comumente de *Full Text*, que é o foco deste trabalho.

*Full Text* (texto completo) é um tipo de dado que difere de cadeia de caractere por causa de seu comprimento e das operações de banco de dados específicas que são aplicáveis a ele. A indexação desses dois tipos também é diferente, enquanto cadeias de caracteres são indexadas por seu valor completo, para *Full Text*, tipos especiais de índices são definidos. Esses tipos especiais de índices permitem obter informações como

a proximidade das palavras ou de frases entre si, bem como termos relacionados contidos no documento (MELTON e EISENBERG, 2001).

Alguns métodos foram adicionados ao UDT *Full Text* para agregar nas pesquisas permitindo a busca por palavras derivadas, como o plural e outros tempos verbais do termo, palavras específicas, sinônimos, palavras com a pronúncia similar. O resultado de uma pesquisa será uma lista de documentos organizada de acordo com a relevância do termo pesquisado (MELTON, 2003).

Para exemplificar o uso do *Full Text*, a Figura 1 exibe o comando para a criação de uma tabela chamada “DVD\_information”, contendo as colunas “stock\_number” e “liner\_notes” que armazenam o identificador, do tipo inteiro, e o documento, do tipo *Full Text*, respectivamente.

**Figura 1.** Criação de tabela para armazenar documentos *Full Text*, de acordo com o padrão SQL/MM.

---

```
CREATE TABLE DVD_information (  
    stock_number INTEGER,  
    liner_notes FULLTEXT)
```

---

Fonte: (MELTON, 2003)

Após a criação e inserção de dados na tabela se torna possível realizar operações com os documentos armazenados, por exemplo, o uso do predicado CONTAINS que pode retornar uma palavra, ou frase, o prefixo da palavra, o termo flexionado do termo pesquisado, como o plural do termo, e um sinônimo. As Figuras 2 e 3 mostram buscas por palavras e frases, respectivamente.

**Figura 2.** Consulta aos documentos armazenados verificando se há o termo “thugs”, de acordo com o padrão SQL/MM.

---

```
SELECT stock_number  
FROM DVD_information  
WHERE liner_notes.CONTAINS (' "thusg" ') = 1
```

---

Fonte: (MELTON, 2003)

**Figura 3.** Consulta aos documentos armazenados verificando se há a frase termo “historical documents”, de acordo com o padrão SQL/MM.

---

```
SELECT stock_number
FROM DVD_information
WHERE liner_notes.CONTAINS ( ' "historical documents" ' ) = 1
```

---

Fonte: (MELTON, 2003)

As consultas das Figuras 2 e 3 retornam a coluna “stock\_number” de todos os documentos que retornarem o valor 1, ou seja, verdadeiro, para o método aplicado na coluna “liner\_notes”.

Com o uso do predicado IN SAME SENTENCE AS é possível consultar documentos que contenham dois termos específicos na mesma frase, como mostra a Figura 4.

**Figura 4.** Consulta aos documentos armazenados verificando se há os termos “attacked” e “crew” na mesma frase, de acordo com o padrão SQL/MM.

---

```
SELECT stock_number
FROM DVD_information
WHERE liner_notes.CONTAINS
      ( ' "attacked" IN SAME SENTENCE AS "crew" ' ) = 1
```

---

Fonte: (MELTON, 2003)

É possível também consultar documentos que possuem palavras com pronúncia similar ao termo pesquisado, por meio do predicado SOUNDS LIKE. A Figura 5 exibe a consulta demonstrando o uso do predicado SOUNDS LIKE em conjunto com o predicado STEMMED FORM OF que encontra palavras derivadas do termo buscado, como por exemplo, “*funnier*” e “*funniest*” sendo o termo buscado “*funny*”. A consulta da Figura 5 retornará documentos que contenham derivados da palavra “*funny*” e termos com pronúncia similar a “*lions*” no mesmo parágrafo.

**Figura 5.** Consulta por derivados de “*funny*” e termos com pronúncia similar a “*lions*” no mesmo parágrafo, de acordo com o padrão SQL/MM.

---

```
SELECT stock_number
FROM DVD_information
WHERE liner_notes.CONTAINS
      ( ' STEMMED FORM OF "funny"
        IN SAME PARAGRAPH AS
        SOUNDS LIKE "lions" ' ) = 1
```

---

Fonte: (MELTON, 2003)

Para consultar documentos por relevância de acordo com o termo pesquisado, é utilizado o predicado RANK. Seu critério é a quantidade de vezes que a palavra aparece no documento. A consulta da Figura 6 exibe documentos com valor de relevância maior que 1.2 em relação ao termo “*carpet*”.

**Figura 6.** Consulta por relevância, de acordo com o padrão SQL/MM.

---

```
SELECT stock_number
FROM DVD_information
WHERE 1.2 < liner_notes.RANK(' "carpet" ')
```

---

Fonte: (MELTON, 2003)

O predicado RANK pode ser utilizado também para retornar todos os documentos em ordem de relevância de acordo com um termo. A Figura 7 mostra uma consulta que retornará todos os documentos em ordem descendente de relevância (*score*) de acordo com o termo “*carpet*”.

**Figura 7.** Consulta por relevância mostrando o valor de relevância, de acordo com o padrão SQL/MM.

---

```
SELECT stock_number, liner_notes.RANK ( ' "carpet" ' ) AS score
FROM DVD_information
WHERE score > 0
ORDER BY score DESCENDING
```

---

Fonte: (MELTON, 2003)

Outro recurso importante para buscas textuais do SQL/MM é o predicado IS ABOUT, que retorna documentos relacionados à palavra ou frase especificada na consulta, como mostra a Figura 8, na qual serão retornados os documentos com conteúdo relacionado a “*science fiction*”.

**Figura 8.** Consulta verificando se há documentos com conteúdos relacionados a *science fiction* de acordo com o padrão SQL/MM.

---

```
SELECT stock_number
FROM DVD_information
WHERE liner_notes.CONTAINS ( ' IS ABOUT "science fiction" ' ) = 1
```

---

Fonte: (MELTON, 2003)

Para buscas nas quais há uma relação entre as palavras ou frases, o uso de dicionário de sinônimos pode ajudar, principalmente quando o objetivo é buscar assuntos específicos ou técnicos (MELTON, 2003).

Importante mencionar que nas consultas podem ser utilizados operadores booleanos como OR, AND e NOT, tornando assim a consulta mais específica para retornar apenas o desejado.

### 3. RECURSOS DO SQL SERVER PARA DADOS *FULL TEXT*

Com o objetivo de encontrar um Sistema de Gerenciamento de Bancos de Dados (SGBD) que possua recursos para o tratamento de dados do tipo *Full Text*, o trabalho de Paiva (2020) apresenta uma análise do DB2, Oracle e SQL Server com suas diferenças, funcionalidades e limitações. As características consideradas na análise foram: tipos de dados suportados, formatos de documentos, idiomas e pesquisas adicionais disponíveis.

Segundo Paiva (2020), os SGBD Oracle e SQL Server apresentam uma maior variedade de funcionalidades e pesquisas adicionais mais amplas para o tratamento de dados *Full Text*, já o DB2 não possui funcionalidades adicionais. Sua avaliação em relação a disponibilidade e usabilidade destaca o SQL Server, dado que ele possui mais materiais de apoio disponíveis pela internet e é oferecido gratuitamente para alunos da UNICAMP.

Tendo em vista os pontos mencionados, o SQL Server foi o SGBD escolhido para ser utilizado neste trabalho. Os tipos de documentos suportados pelo SQL Server são: PDF, HTML e XML. O mecanismo que permite consultas de texto completo do SQL Server é chamado *Full Text Search*, um componente opcional que não é instalado junto com o SQL Server por padrão. Caso o SQL Server já esteja instalado, o software de ver reinstalado em conjunto do recurso *Full Text Search*, é preciso também, o download do recurso chamado banco de dados de estatísticas semânticas de idioma, recurso requerido para realizar buscas semânticas que contém modelos de idioma estatísticos (MICROSOFT, 2021).

Para consultas a documentos do tipo HTML, é requerido o uso de um filtro HTML, para verificar se o filtro já se encontra instalado basta executar o comando presente no Apêndice B. Já para consultas em documentos do tipo PDF é necessário a instalação de um *plug-in* da Adobe, dado que nativamente, o SQL Server dispõe de suporte apenas para formatos de documentos proprietários da Microsoft. Após a instalação e configuração do *plug-in*, é possível realizar as consultas a documentos do tipo PDF, detalhado no Apêndice B. Além de consultas em PDF, XML e HTML é possível realizar consultas *Full Text* em dados baseados em caracteres armazenados diretamente no banco de dados.

As consultas do mecanismo *Full Text Search* executam as buscas em índices *Full Text*. O índice corresponde a coluna que armazena o conteúdo *Full-Text*. Essa coluna permite os seguintes tipos de dados: *char*, *varchar*, *nchar*, *nvarchar*, *text*, *ntext*, *image*, *xml* ou *varbinary*. Com o intuito de simplificar os exemplos foi utilizado *varchar* para realizar as consultas (MICROSOFT, 2021).

O mecanismo de busca *Full Text Search* permite consultas a uma ou mais palavras específicas, formas flexionadas de um termo, palavras próximas de outras palavras específicas e sinônimos de um termo. Para realizar as consultas mencionadas, é necessário realizar a criação de um catálogo de índice *Full Text* e a criação de índices *Full Text*.

Para exemplificar, o uso do tipo *Full Text*, a Figura 9 mostra a criação da tabela “artigo” contendo as seguintes colunas: “id\_artigo”, “titulo” e “descricao”.

**Figura 9.** Criação de tabela no SQL Server

---

```
CREATE TABLE artigo (  
    id_artigo INT,  
    titulo VARCHAR(100),  
    descricao VARCHAR (200),  
    PRIMARY KEY(id_artigo)  
)
```

---

Fonte: Adaptado de Microsoft (2021)

As Figuras 10 e 11 exibem, respectivamente, os *scripts* para criação de catálogo e de índice para a tabela “artigo”. Antes de realizar a criação de um índice *Full Text* é necessário criar um catálogo *Full Text*, que é um agrupamento de lógico para um grupo de índices *Full Text* (MICROSOFT, 2021).

**Figura 10.** Criação de catálogo no SQL Server.

---

```
CREATE FULLTEXT CATALOG ftCatalogo
```

---

Fonte: Adaptado de Microsoft (2021)

**Figura 11.** Criação de índice no SQL Server.

---

```
CREATE FULLTEXT INDEX ON artigo(descricao)
```

---

Fonte: Adaptado de Microsoft (2021)

Após a criação do catálogo e do índice é possível realizar as consultas de texto *Full Text*. Para isso o SQL Server dispõe dos predicados CONTAINS e FREETEXT e as funções CONTAINSTABLE e FREETEXTTABLE, ambas retornam o valor de classificação de relevância (RANK) e a chave de texto completo (KEY), identificador exclusivo de cada linha. Os valores retornados na coluna RANK indicam uma ordem relativa de relevância das linhas do conjunto de resultado, que varia de 0 a 1000, sendo que o maior valor indica maior relevância. Os valores reais não são relevantes e podem ser diferentes a cada execução, apontam apenas a ordem relativa de relevância. O algoritmo padrão utilizado para a classificação é chamado Jaccard (MICROSOFT, 2021).

O predicado CONTAINS busca correspondências exatas ou difusas para palavras ou frases, palavras a uma determinada distância entre si ou correspondências ponderadas. É utilizado na cláusula WHERE de uma instrução SELECT. Importante mencionar que as consultas não diferem letras maiúsculas de minúsculas.

A Figura 12 exibe uma busca de todas as descrições de artigos que contenham o termo “SQL”.

**Figura 12.** Consulta por palavra específica utilizando o SQL Server.

---

```
SELECT titulo, descricao
FROM artigo
WHERE CONTAINS (descricao, 'SQL')
```

---

Fonte: Adaptado de Microsoft (2021)

É possível realizar consultas por frases utilizando o CONTAINS como mostra a Figura 13.

**Figura 13.** Consulta por frase utilizando o SQL Server.

---

```
SELECT titulo, descricao
FROM artigo
WHERE CONTAINS (descricao, ' "Full Text Search" ')
```

---

Fonte: Adaptado de Microsoft (2021)

Para realizar consultas por palavras que começam com um determinado texto o predicado CONTAINS também pode ser utilizado. Após prefixo deve se utilizar asterisco (\*). A Figura 14 mostra consulta por palavras que iniciam com “inter”.

**Figura 14.** Consulta por prefixo utilizando o SQL Server.

---

```
SELECT titulo, descricao
FROM artigo
WHERE CONTAINS (descricao, ' "inter*" ')
```

---

Fonte: Adaptado de Microsoft (2021)

A função CONTAINSTABLE busca as mesmas correspondências que o CONTAINS, entretanto ao utilizá-la é possível exibir uma lista de importância com as linhas retornadas. É utilizada na cláusula FROM já que retorna uma tabela temporária. A Figura 15 mostra um exemplo de seu uso.

**Figura 15.** Consulta por relevância de um termo utilizando o SQL Server.

---

```
SELECT FT_TBL.titulo, FT_TBL.descricao, KEY_TBL.RANK
FROM artigo AS FT_TBL
INNER JOIN CONTAINSTABLE (artigo, descricao, ' "Full Text" ') AS
KEY_TBL
ON FT_TBL.id_artigo = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
```

---

Fonte: Adaptado de Microsoft (2021)

É possível realizar a pesquisa para todas colunas indexadas de texto completo, se houver, para isso basta substituir o segundo argumento da função por asterisco (\*).

Para consultar as descrições pelo seu significado, o predicado FREETEXT pode ser utilizado na cláusula WHERE. Quando utilizado, o mecanismo de busca atribui um peso para cada termo, e então, localiza as correspondências para fragmentos das palavras e formas flexionadas como mostra a Figura 16. Para essa consulta documentos com os termos “technologies” e “technologic” serão retornados, por exemplo. E para obter uma lista por ordem de relevância a função FREETEXTTABLE pode ser utilizada como mostra a Figura 17. A relevâncias dos resultados é baseada na fórmula de classificação OKAPI BM25 (MICROSOFT, 2021).

**Figura 16.** Consulta por correspondências difusas de um termo utilizando SQL Server.

---

```
SELECT titulo, descricao
FROM artigo
WHERE FREETEXT (descricao, 'technology')
```

---

Fonte: Adaptado de Microsoft (2021)

**Figura 17.** Consulta por correspondências difusas de um termo utilizando SQL Server.

```
SELECT FT_TBL.titulo, FT_TBL.descricao, KEY_TBL.RANK
FROM artigo AS FT_TBL
      INNER JOIN FREETEXTTABLE (artigo, descricao, 'technology') AS
KEY_TBL
      ON FT_TBL.id_artigo = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
```

Fonte: Adaptado de Microsoft (2021)

Para realizar buscas que identificam sinônimos é necessária a configuração do dicionário de sinônimos que é instalado juntamente do recurso *Full Text Search*. Para isso basta localizar o arquivo XML na pasta de instalação do SQL Server e definir os sinônimos dos termos que deve ser realizada a correspondência. O processo detalhado se encontra no apêndice A. Para exemplificar, a Figura 18 exibe o conteúdo do arquivo para sinônimos do termo “search”, dessa forma, consultas pelo termo mencionado retornam as linhas que contenham as palavras “check”, “investigate” e “examine”.

**Figura 18.** Dicionário de sinônimos.

```
<XML ID="Microsoft Search Thesaurus">
  <thesaurus xmlns="x-schema:tsSchema.xml">
    <diacritics_sensitive>0</diacritics_sensitive>
    <expansion>
      <sub> search </sub>
      <sub> check </sub>
      <sub> investigate </sub>
      <sub> examine </sub>
    </expansion>
  </thesaurus>
</XML>
```

Fonte: Desenvolvido pelo autor

É possível atribuir um valor de importância para um termo e realizar uma consulta por termo ponderado com o uso da função *CONTAINSTABLE* em conjunto com o operador *ISABOUT*, que permite que usuário especifique um valor de 0.0 (menor relevância) a 1.0 (maior relevância) para um termo ou frase. O valor de relevância é calculado a partir do algoritmo de Jaccard (MICROSOFT, 2021). A Figura 19 mostra uma consulta por correspondências dos termos “*Full Text*” e “*Search*”.

**Figura 19.** Consulta a termos ponderados no SQL Server.

---

```
SELECT FT_TBL.titulo, FT_TBL.descricao, KEY_TBL.RANK
FROM artigo AS FT_TBL
    INNER JOIN CONTAINSTABLE(artigo, descricao,
        'ISABOUT ("Full Text" WEIGHT (.1),
            "Search" WEIGHT (.1) ) AS KEY_TBL
    ON FT_TBL.id_artigo = KEY_TBL.[KEY]
WHERE FT_TBL.id_artigo > 1
ORDER BY KEY_TBL.RANK DESC
```

---

Fonte: Adaptado de Microsoft (2021)

O SQL Server também conta com as funções semânticas SEMANTICKEYPHRASETABLE, SEMANTICSIMILARITYDETAILSTABLE e SEMANTICSIMILARITYTABLE que possibilitam consultas semânticas. Para realizar as consultas semânticas é necessário realizar a criação de um índice semântico para a coluna onde as buscas serão realizadas, no caso do exemplo apresentado na Figura 19, a coluna “descricao”. É necessário definir também o código do idioma, para português – Brasil, o identificador é “1046”. A Figura 20 demonstra a criação do índice semântico.

**Figura 20.** Criação de um índice semântico na coluna “descrição” no SQL Server

---

```
CREATE FULLTEXT INDEX ON artigo
    (descricao
        Language 1046 Statical_Semantics
    )
KEY INDEX id_artigo
WITH STOPLIST =SYSTEM
```

---

Fonte: Adaptado de Microsoft (2021)

A primeira função semântica mencionada, SEMANTICKEYPHRASETABLE, utilizada na cláusula FROM, retorna uma tabela com as linhas que apresentem a palavra-chave especificada. A consulta da Figura 21 exibe a consulta por linhas de documentos que mencionem “semantic” por ordem de relevância. Importante frisar que apenas palavras podem ser consultadas no uso das funções semânticas, dado que somente palavras isoladas são indexadas (MICROSOFT, 2021).

**Figura 21.** Consulta a linhas de documentos que apresentem um termo específico no SQL Server.

---

```
SELECT s.keyphrase, s.SCORE, d.id_artigo, d.descricao,  
FROM SEMANTICKEYPHRASETABLE (artigo, decrial) AS s  
INNER JOIN artigo AS d ON  
s.document_key = d.id_artigo  
WHERE s.keyphrase = 'semantic'  
ORDER BY s.SCORE DESC
```

---

Fonte: Adaptado de Microsoft (2021)

O predicado SEMANTICSIMILARITYDETAILSTABLE, utilizado na cláusula FROM, busca a mesma correspondência em dois documentos diferentes e seus respectivos SCORE. Por exemplo, é possível buscar quais termos dois documentos possuem em comum como mostra a Figura 22.

**Figura 22.** Consulta por termos que os documentos de ID 1 e 2 possuem em comum, com SCORE superior à 0.6.

---

```
SELECT s.keyphrase, s.SCORE  
FROM SEMANTICSIMILARITYDETAILSTABLE(artigo, descricao,1,  
descricao, 2) AS s  
WHERE S.SCORE > 0.6  
ORDER BY s.SCORE DESC
```

---

Fonte: Adaptado de Microsoft (2021)

Com o predicado SEMANTICSIMILARITYTABLE, utilizado na cláusula FROM, é possível buscar documentos que possuam conteúdos semanticamente correlatos nas colunas especificadas, como mostra a Figura 23 que exhibe uma comparação de uma descrição com as demais, apresentando um valor de semelhança para cada descrição, o SCORE.

**Figura 23.** Consulta que exhibe o grau de similaridade de um documento com o restante no SQL Server.

---

```
SELECT s.matched_document_key, s.SCORE, d.descricao  
FROM SEMANTICSIMILARITYTABLE(artigo, descricao, 1) AS s  
INNER JOIN artigo AS d  
ON s.matched_document_key = d.id_artigo  
ORDER BY s.SCORE DESC
```

---

Fonte: Adaptado de Microsoft (2021)

#### **4. ESTUDO DE CASO**

O presente trabalho propõe a utilização dos recursos de pesquisa semântica em conjunto com os recursos para consultas a texto completo disponíveis no SQL Server para realizar a recuperação de textos por meio da similaridade de conceitos.

Para o estudo de caso, um banco de dados foi criado para armazenar comentários de alunos realizados em um blog utilizado em uma disciplina de uma faculdade, na qual o professor contabiliza manualmente quais conceitos foram mencionados em cada comentário relacionado a um determinado tópico. A extração realizada no blog contém 4 comentários e todos os conceitos mencionados em aula sobre um tópico abordado. A aplicação dos recursos estudados deve ser capaz de identificar os conceitos nos comentários dos alunos e determinar um valor de relevância em relação a cada conceito, de modo a obter um valor de relevância final do comentário. Quanto maior o valor, maior o grau de similaridade do comentário com o tópico abordado em aula.

Os seguintes comentários foram armazenados em uma tabela denominada “Resposta”, que contém as colunas “Id” e “Conteúdo”, que são referentes ao identificador do comentário e o conteúdo, respectivamente. A estrutura detalhada da criação da tabela “Resposta” e a inserção dos dados se encontra no Apêndice A.

##### **Comentário 1**

Olá Luiz! Boa questão. Nesse âmbito, a expectativa de lucro, por parte dos stakeholders/sócios afeta sim o preço do produto, pois, para que exista o lucro, a soma das despesas e dos custos não deve ser maior que o valor da receita de vendas. Esse retorno é importante para os stakeholders pois demonstra que o produto em que ele investiu teve seu market share. A importância dos stakeholders, assim como os colegas disseram acima, varia com a situação, sobre quais eles podem fazer diferentes papéis.

##### **Comentário 2**

Olá Pedro, em relação a sua dúvida do CMAPS, o sentido do fluxo de capital ser negativo é que existe fluxo de capital negativo e temos que considera-lo. No fluxo de capital existem dois fluxos: o de entrada e saída. Fluxo de entrada vem das receitas de vendas e o de saída dos custos e despesas. Diante disso a soma dos custos e despesas

não pode ultrapassar o valor da Receita de vendas, pois é necessário que a empresa tenha LUCRO, para contribuir com impostos. Para entregarmos valor precisamos ser atrativos, e para isso tem que ter uma expectativa de lucro real, ou seja, FLUXO DE CAPITAL POSITIVO (receita de vendas maior que Despesas + custos).

### **Comentário 3**

Bom dia, eu fiquei com uma dúvida em relação a possíveis sócios da empresa que que auxiliariam com o Capital Social. O preço do produto deve ser comparado com o preço do concorrente e é composto por lucro, custos, despesas e impostos porém ainda devemos nos atentar para a expectativa de lucro dos sócios e ainda gerar valor para empresa. Minha dúvida é se a expectativa de lucro dos sócios afeta diretamente o preço do produto, pois uma vez que investiram na empresa esperam um retorno válido, o que pode acabar afetando a atratividade de mercado do nosso produto e diminuir nosso diferencial competitivo nesse quesito. Sendo assim se a empresa tiver a possibilidade de iniciar sem a ajuda de um sócio seria melhor? ou é melhor tentar balancear a expectativa dos sócios com as necessidades da empresa, mesmo se a empresa tiver condições para iniciar sem a necessidade de uma sociedade?

### **Comentário 4**

Olá pessoal, achei uma boa ideia o ponto levantado pelo Victor e comentado pelo Arthur. Se a Hexágono buscar por membros que realmente são bons em publicidade, isso levará um aumento da expectativa de lucro real, o que, por sua vez, aumenta a atratividade do mercado. Isso ocorre porque o fluxo de capital (Receita de vendas > Despesas + Custos) será positivo e superior aos outros tipos de investimentos, uma vez que estaríamos diminuindo os custos. Ressalta-se que isso só será possível se os membros contratados forem empenhados e saberem trabalhar com publicidade, caso contrário voltamos ao estágio inicial. Sobre a proposta de dar bolsas para estes membros, citada pelo Arthur, também concordo que seria um ótimo cenário para ambas as partes. Alguém tem alguma outra ideia para contratar estes membros?

Os conceitos usados nas consultas foram: Atratividade do Mercado, Capital Econômico, Capital Intelectual, Capital Social, Custo de Produção, Despesas,

Expectativa de Lucro, Fluxo de Capital, Inovação, Investimento, Lucro, Preço do produto, Receita de vendas, Stakeholders, Sócio, Tributos e Valor para a Empresa.

Conforme citado no Capítulo 3, após a configuração do SQL Server e o armazenamento de dados *Full Text*, é possível realizar consultas utilizando o mecanismo *Full Text Search*. A partir das consultas, foi analisada a efetividade dos recursos estudados, aplicados à recuperação de textos por meio da similaridade de conceitos.

Para recuperar termos ou frases, pode-se utilizar os predicados CONTAINS ou FREETEXT. Entretanto o primeiro retorna apenas correspondências exatas, enquanto o segundo retorna linhas que apresentam termos flexionados do termo/frase informado. As Figuras 24 e 25 exibem o uso dos predicados CONTAINS e FREETEXT, respectivamente.

**Figura 24.** Consulta por conceitos utilizando CONTAINS

---

```
SELECT *  
FROM Resposta  
WHERE CONTAINS (Conteudo, ' "receita de venda" ')
```

---

Fonte: Desenvolvido pelo autor

**Figura 25.** Consulta por conceitos utilizando FREETEXT

---

```
SELECT *  
FROM Resposta  
WHERE FREETEXT (Conteudo, 'receita de venda')
```

---

Fonte: Desenvolvido pelo autor

A consulta utilizando CONTAINS (Figura 24) não retorna nenhum resultado. Dado que nos comentários armazenados encontra-se o conceito “receita de vendas”, a condição de correspondência exata não é satisfeita. Ao utilizar o predicado FREETEXT, a consulta da Figura 25 retorna 3 linhas, que correspondem aos comentários que apresentam os conceitos “receita de vendas” e “receitas de vendas”, similares ao conceito especificado na consulta, como mostra a Figura 26.

**Figura 26.** Resultados obtidos na consulta da Figura 25

	Id	Conteudo
1	1	Olá Luiz! Boa questão. Nesse âmbito, a expectativa ...
2	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...
3	4	Olá pessoal, achei uma boa ideia o ponto levantado ...

Fonte: Desenvolvido pelo autor

Para casos em que há necessidade de expandir a busca por termos similares, o predicado FREETEXT se mostra mais efetivo. Para realizar esse tipo de busca e exibir a relevância dos resultados de acordo com o conceito especificado, pode-se utilizar a função FREETEXTTABLE, como mostra a Figura 27.

**Figura 27.** Consulta por conceitos utilizando FREETEXTTABLE

---

```
SELECT *
FROM Resposta AS FT_TBL
      INNER JOIN FREETEXTTABLE(Resposta, Conteudo,
      'receita de venda') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK
```

---

Fonte: Desenvolvido pelo autor

A Figura 28 exibe o resultado da consulta da Figura 27, onde o comentário 2 apresenta maior valor de relevância. Isso ocorre devido ao seu conteúdo apresentar conceitos similares ao especificado três vezes, enquanto os comentários 1 e 4 apresentam apenas uma vez.

**Figura 28.** Resultados obtidos na consulta da Figura 27

	Id	Conteudo	KEY	RANK
1	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...	2	687
2	1	Olá Luiz! Boa questão. Nesse âmbito, a expectativa ...	1	423
3	4	Olá pessoal, achei uma boa ideia o ponto levantado ...	4	359

Fonte: Desenvolvido pelo autor

Ao realizar a consulta por relevância utilizando a função CONTAINSTABLE, exibida na Figura 29, nenhuma linha é retornada dado que não há correspondência exata com o conceito “receita de venda”.

**Figura 29.** Consulta por relevância utilizando CONTAINSTABLE

---

```
SELECT *
FROM Resposta AS FT_TBL
      INNER JOIN CONTAINSTABLE(Resposta, Conteudo, ' "receita
de venda" ' ) AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
```

---

Fonte: Desenvolvido pelo autor

Como abordado no Capítulo 3, para expandir as buscas é possível configurar um dicionário de sinônimos. A Figura 30 apresenta o arquivo de configuração do dicionário, considerando os termos relevantes para o estudo caso. O elemento “diacritics\_sensitive” define se o dicionário é sensível ou não a acentos, contendo apenas um inteiro, sendo que um (1) é usado para que haja distinção de acentos e zero (0) para que não haja (MICROSOFT, 2022).

**Figura 30.** Dicionário de sinônimos.

---

```
<XML ID="Microsoft Search Thesaurus">
  <thesaurus xmlns="x-schema:tsSchema.xml">
    <diacritics_sensitive>0</diacritics_sensitive>
    <expansion>
      <sub>tributos</sub>
      <sub>impostos</sub>
      <sub>taxas</sub>
    </expansion>
    <expansion>
      <sub>atratividade</sub>
      <sub>atrativos</sub>
    </expansion>
    <expansion>
      <sub>investimento</sub>
      <sub>capital</sub>
    </expansion>
    <expansion>
      <sub>preço</sub>
      <sub>custos</sub>
      <sub>valor</sub>
      <sub>despesas</sub>
    </expansion>
    <expansion>
      <sub>vendas</sub>
      <sub>lucro</sub>
      <sub>receita</sub>
      <sub>produto</sub>
    </expansion>
    <expansion>
      <sub>sócio</sub>
      <sub>membros</sub>
    </expansion>
    <expansion>
      <sub>empresa</sub>
      <sub>sociedade</sub>
    </expansion>
  </thesaurus>
</XML>
```

---

Fonte: Desenvolvido pelo autor

Com o intuito de explanar a configuração do dicionário, a Tabela 1 exibe os tópicos e os sinônimos correspondentes de cada termo. As palavras destacadas de verde estão presentes no texto extraído, já as palavras destacadas de vermelho não estão presentes. Apenas os termos realçados em verde foram adicionados no dicionário de sinônimos,

desse modo, evitando um maior tempo de processamento indevido. O processo de configuração do dicionário detalhado se encontra no apêndice A.

**Tabela 1.** Conceitos e sinônimos de cada termo

Conceito	1º termo	2º termo
Atratividade do Mercado	atrativo	
Capital Econômico	Patrimônio, riqueza, tesouro, recursos	Financeiro, financeira
Capital Intelectual	Patrimônio, riqueza, tesouro, recursos	Intelectivo, mental, racional
Capital Social	Patrimônio, riqueza, tesouro, recursos	Coletivo, comunitário, público, geral, grupal
Custo de Produção	Gasto, dispêndio, desembolso, despesa, consumo	fabricação, fabrico, feitura, manufatura, manufatura, construção, montagem, confecção, formação, geração.
Despesas	custos, expensa, gasto, importe, preço, sumpto, suntu.	
Expectativa de Lucro	chance, promessa, perspectiva, viabilidade, contingência.	rendimento, ganho, juro, renda, produto, provento, dividendo, emolumento, granjeio, logro.
Fluxo de Capital	deslocamento, movimentação, movimento, passagem, trânsito.	Patrimônio, riqueza, tesouro, recursos
Inovação	Novidade, nova, revolução	
Investimento	aplicação, inversão, capital, dinheiro, patrimônio, riqueza, tesouro, bens, posses, haveres, meios, pecúlio.	
Lucro	rendimento, ganho, juro, renda, produto, provento, dividendo, emolumento, granjeio, logro.	
Preço do produto	custo, importe, montante, quantia, valia, valor.	artigo, bem, mercadoria, mercancia, gênero, item, objeto, compra, peça, volume, artefato, veniaga.
Receita de vendas	ganho, lucro, renda, faturação, faturamento, fêria, produto, rendimento, arrecadação.	fêria, apurado, movimento, renda, ganho, lucro, receita, produto, faturamento, faturação.
Stakeholders	Termos não exatos	
Sócio	associado, membro, societário.	
Tributos	impostos, contribuições, contribuições, taxas.	
Valor para a Empresa	preço, porte, quantia, número, quantidade, montante, monta, custo, cifra, cotação, grandeza, medida.	companhia, firma, casa, negócio, sociedade, entidade, estabelecimento, instituição, organização, empregador, parceria, corporação.

Fonte: Desenvolvido pelo autor

Após a configuração do dicionário, buscas utilizando FREETEXT e FREETEXTABLE, retornam linhas que apresentam sinônimos dos termos buscados, como mostra a Figura 31. Ao realizar a consulta pelo termo “tributos” as linhas 2 e 3 são exibidas no resultado (Figura 32) dado que apresentam o termo “impostos”, configurado como sinônimo de “tributos”.

**Figura 31.** Consulta por correspondências difusas de um termo utilizando FREETEXT.

---

```
SELECT titulo, descricao
FROM Resposta
WHERE FREETEXT (Conteudo,'tributos')
```

---

Fonte: Adaptado de Microsoft (2021)

**Figura 32.** Resultados obtidos na consulta da Figura 31

	Id	Conteudo
1	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...
2	3	Bom dia, eu fiquei com uma dúvida em relação a po...

Fonte: Desenvolvido pelo autor

As consultas utilizando CONTAINS e CONTAINSTABLE também podem realizar correspondências com o dicionário, basta especificar a cláusula FORMSOF THESAURUS, como mostram as Figuras 33 e 34.

**Figura 33.** Consulta por sinônimos utilizando CONTAINS

---

```
SELECT *
FROM Resposta
WHERE CONTAINS(Conteudo,'FORMSOF (THESAURUS,"tributos")')
```

---

Fonte: Adaptado de Microsoft (2021)

**Figura 34.** Consulta por sinônimos utilizando CONTAINSTABLE.

---

```
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE (Resposta, Conteudo, 'FORMSOF
    (THESAURUS, "tributos") ') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
```

---

Fonte: Desenvolvido pelo autor

As Figuras 35 e 36 exibem os resultados das buscas por sinônimos do termo “tributos” utilizando CONTAINS e CONTAINSTABLE, respectivamente.

**Figura 35.** Resultados obtidos na consulta da Figura 33

	Id	Conteudo
1	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...
2	3	Bom dia, eu fiquei com uma dúvida em relação a po...

Fonte: Desenvolvido pelo autor

**Figura 36.** Resultados obtidos na consulta da Figura 34

	Id	Conteudo	KEY	RANK
1	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...	2	4
2	3	Bom dia, eu fiquei com uma dúvida em relação a po...	3	4

Fonte: Desenvolvido pelo autor

Além dos recursos de *Full Text Search*, o SQL Server dispõe das funções semânticas SEMANTICKEYPHRASETABLE, SEMANTICSIMILARITYTABLE e SEMANTICSIMILARITYDETAILSTABLE. A função SEMANTICKEYPHRASETABLE apresenta recursos que possibilitam identificar palavras-chaves em textos e calcula o valor de relevância denominado SCORE, no intervalo de 0,0 a 1,0. O retorno da função SEMANTICKEYPHRASETABLE é uma tabela com quatro colunas, sendo elas: “column\_id”, “document\_key”, “keyphrase” e “score”.

Conforme abordado no Capítulo 3, a pesquisa semântica não é capaz de buscar por frases, apenas palavras. Com o objetivo de buscar os conceitos e não apenas palavras-chaves, é possível utilizar as funções semânticas em conjunto com recursos *Full Text*. A Figura 37 exhibe a consulta por comentários que apresentam a palavra-chave “expectativa” e o termo “lucro”.

**Figura 37.** Consulta por comentários com os termos “expectativa” e “lucro”

---

```

SELECT s.keyphrase, s.SCORE, r.Id, r.Conteudo
FROM SEMANTICKEYPHRASETABLE(Resposta, Conteudo) AS s
INNER JOIN Resposta AS r
ON s.document_key = r.Id
WHERE s.keyphrase = 'expectativa' AND CONTAINS(Conteudo,'lucro')
ORDER BY s.SCORE DESC

```

---

Fonte: Desenvolvido pelo autor

A Figura 38 exibe o resultado da consulta da Figura 37, na qual o comentário 3 apresentou maior relevância, dado que apresenta os termos especificados mais vezes quando comparado ao restante dos comentários. As colunas “keyphrase”, “score” mostram a palavra-chave identificada na busca e a relevância de cada comentário, respectivamente.

**Figura 38.** Resultados obtidos na consulta da Figura 37

	keyphrase	SCORE	Id	Conteudo
1	expectativa	0,4605258	3	Bom dia, eu fiquei com uma dúvida em relação a po...
2	expectativa	0,1791788	1	Olá Luiz! Boa questão. Nesse âmbito, a expectativa ...
3	expectativa	0,1791788	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...
4	expectativa	0,1677495	4	Olá pessoal, achei uma boa ideia o ponto levantado ...

Fonte: Desenvolvido pelo autor

Para filtrar ainda mais as consultas, é possível utilizar o operador NEAR em conjunto com CONTAINS ou CONTAINSTABLE, a fim de consultar termos próximos e especificar a distância máxima entre as palavras. A Figura 39 mostra a consulta por comentários que apresentam os termos “fluxo” e “capital” com uma distância máxima de até 2 palavras entre eles.

**Figura 39.** Consulta por comentários com os termos “fluxo” e “capital” próximos um ao outro.

---

```
SELECT *
FROM Resposta
WHERE CONTAINS (Conteudo,'NEAR((fluxo, capital),2)')
```

---

Fonte: Desenvolvido pelo autor

A Figura 40 exibe o resultado da consulta da Figura 39, que retornou os comentários 2 e 4, que apresentam o conceito “Fluxo de capital”.

**Figura 40.** Resultados obtidos na consulta da Figura 39

	Id	Conteudo
1	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...
2	4	Olá pessoal, achei uma boa ideia o ponto levantado ...

Fonte: Desenvolvido pelo autor

A Figura 41 exibe a consulta por comentários que apresentem os termos “receita” e “vendas” próximo um ao outro utilizando CONTAINSTABLE.

**Figura 41.** Consulta por comentários com os termos “receita” e “vendas” próximos um ao outro.

```
SELECT *  
FROM Resposta AS FT_TBL  
      INNER JOIN CONTAINSTABLE(Resposta, Conteudo,'NEAR((receita,  
      vendas),2)') AS KEY_TBL  
ON FT_TBL.Id = KEY_TBL.[KEY]  
ORDER BY KEY_TBL.RANK DESC
```

Fonte: Desenvolvido pelo autor

A Figura 42 exibe o resultado da consulta da Figura 41, que retornou os comentários 2, 4 e 1. Dado que CONTAINSTABLE busca correspondências exatas, a busca não contabiliza o conceito “receitas de vendas” ou outras formas flexionadas do conceito.

**Figura 42.** Resultados obtidos na consulta da Figura 41

	Id	Conteudo	KEY	RANK
1	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...	2	9
2	4	Olá pessoal, achei uma boa ideia o ponto levantado ...	4	4
3	1	Olá Luiz! Boa questão. Nesse âmbito, a expectativa ...	1	4

Fonte: Desenvolvido pelo autor

A Figura 43 exibe o resultado da consulta pelos termos “receitas” e “vendas” próximos um do outro. Apenas a resposta 2 é retornada, dado que é a única que tem correspondência exata para os termos “receitas” e “vendas”, com a condição de distância máxima de duas palavras entre os termos.

**Figura 43.** Resultados obtidos na consulta por “receitas” e “vendas”

	Id	Conteudo	KEY	RANK
1	2	Olá Pedro, em relação a sua dúvida do CMAPS, o s...	2	4

Fonte: Desenvolvido pelo autor

Importante destacar algumas limitações dos recursos apresentados, as funções semânticas não permitem a busca de frases, apenas palavras. As consultas com FREETEXT ou FREETEXTTABLE não podem ser combinadas com WEIGHT, FORMSOF ou NEAR. O operador AND, se utilizado em conjunto com FREETEXT ou FREETEXTTABLE, é desconsiderado, dado que é tratado como palavra irrelevante (*stop word*). O operador NEAR não pode ser utilizado em conjunto com ISABOUT ou FORMSOF (MICROSOFT, 2021).

## 4.1 Proposta de solução

A partir dos recursos estudados, duas consultas distintas foram elaboradas com o propósito de recuperar os comentários do blog por meio dos conceitos propostos. Partiu-se do pressuposto que a consulta seja capaz recuperar textos a partir de uma frase/conceito e que a similaridade de termos fosse verificada, considerando sinônimos. As consultas foram enumeradas a fim de auxiliar para o entendimento da análise realizada. Com o intuito de tornar a análise mais dinâmica, apenas os resultados obtidos com quatro conceitos foram explicitados, sendo estes: Lucro, Preço do produto, Receita de vendas e Tributos. O script completo, considerando todos os conceitos, pode ser encontrado no Apêndice C.

As consultas elaboradas exibem nas condições de busca a palavra “conceito”. Dessa forma ao realizar a busca por um conceito específico é necessário substituir “conceito” pelo conceito a ser buscado. Por exemplo, para recuperar os comentários que apresentem o conceito “Atratividade do Mercado”, deve-se modificar a “conceito” para “Atratividade do Mercado.

A Figura 44 exibe a Consulta 1, que utiliza a função FREETEXTTABLE, considerando todos os sinônimos dos termos presentes na consulta e a forma flexionada dos termos.

**Figura 44.** Consulta utilizando FREETEXTTABLE

---

```
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN FREETEXTTABLE(Resposta, *, 'conceito') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
```

---

Fonte: Desenvolvido pelo autor

A Figura 45 exibe a Consulta 2, que utiliza a função CONTAINSTABLE em conjunto com predicado FORMSOF, que realiza a correspondência dos termos da consulta com o dicionário de sinônimos configurado.

**Figura 45.** Consulta utilizando CONTAINSTABLE e FORMSOF

```
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "conceito") ') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
```

Fonte: Desenvolvido pelo autor

As consultas 1 e 2 foram realizadas para os cinco conceitos mencionados. Cada consulta apresenta o identificador do comentário e seu valor de relevância (RANK). Conforme abordado no Capítulo 3, os valores reais de RANK não são relevantes, indicam apenas a ordem relativa de relevância. Importante ressaltar que os valores de RANK para cada consulta são distintos em razão dos diferentes algoritmos utilizados pelo SQL Server (MICROSOFT, 2021).

A Figura 46 exibe os resultados das consultas para o conceito: “Lucro”. Os resultados exibem a mesma ordem de relevância para os comentários armazenados. O comentário 2 apresenta mais vezes o termo “lucro” e sinônimos em comparação com o restante dos comentários. Alguns termos configurados como sinônimos de “lucro”, sendo estes: “receita” e “vendas” também estão presentes no comentário 2. Que ocorrem com menor frequência no comentário 1.

**Figura 46.** Resultados obtidos na consulta pelo conceito “Lucro”

CONSULTA 1		CONSULTA 2	
KEY	RANK	KEY	RANK
2	837	2	16
1	815	1	13
3	770	3	12
4	627	4	6

Fonte: Desenvolvido pelo autor

A Figura 47 exibe os resultados das consultas executadas para o conceito “Preço do produto”. A relevância relativa dos resultados é similar. É possível observar que a Consulta 1, que utiliza a função FREETEXTTABLE retorna os comentários mesmo quando seus valores de RANK são zero, enquanto a Consulta 2, que utiliza a função CONTAINSTABLE não retornar comentários caso a relevância seja zero.

**Figura 47.** Resultados obtidos na consulta pelo conceito “Preço do produto”

CONSULTA 1		CONSULTA 2	
KEY	RANK	KEY	RANK
3	528	3	12
1	423	1	6
2	0		
4	0		

Fonte: Desenvolvido pelo autor

A Figura 48 exibe os resultados para as consultas pelo conceito “Receita de vendas”, que mostra resultados similares aos realizados para o conceito “Preço do produto”. A diferença de ordem de relevância da Consulta 1 é justificada, dado que a função FREETEXTTABLE, considera os termos flexionados, considerando assim a ocorrência da frase “receitas de vendas”, entre outras formas flexionadas, não avaliada na Consulta 2.

**Figura 48.** Resultados obtidos na consulta pelo conceito “Receita de vendas”

CONSULTA 1		CONSULTA 2	
KEY	RANK	KEY	RANK
2	687	2	13
1	423	4	6
4	359	1	6
3	0		

Fonte: Desenvolvido pelo autor

A Figura 49 exibe os resultados obtidos das consultas para o conceito “Tributos”. Apesar de não haver o termo “tributos” em nenhum dos comentários, há correspondências do seu sinônimo, “impostos” nos comentários 2 e 3. O resultado da Consulta 2 apresenta a mesma relevância para os comentários 2 e 3, tal fato é justificado devido aos diferentes algoritmos utilizados em cada consulta.

**Figura 49.** Resultados obtidos na consulta pelo conceito “Tributos”

CONSULTA 1		CONSULTA 2	
KEY	RANK	KEY	RANK
2	423	2	4
3	359	3	4

Fonte: Desenvolvido pelo autor

Se necessário, é possível realizar consultas utilizando mais de um conceito com a função CONTAINSTABLE. A Figura 50 exibe a consulta por comentários que mencionam os conceitos “Receita de vendas” e “Expectativa de Lucro”.

**Figura 50.** Consulta utilizando dois conceitos com CONTAINSTABLE

---

```
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "receita de vendas")
        AND FORMSOF(THESAURUS, "expectativa de lucro")') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
```

---

Fonte: Desenvolvido pelo autor

A Figura 51 mostra o resultado da consulta da Figura 50. Apenas comentários que exibem os dois conceitos são retornados, em função da utilização do operador AND. É possível realizar a mesma consulta utilizando o operador OR, que retorna comentários que apresentem ao menos um dos conceitos mencionados na condição de busca, conforme exibe a Figura 52.

**Figura 51.** Resultados obtidos na consulta da Figura 50

KEY	RANK
1	6
2	6
4	6

Fonte: Desenvolvido pelo autor

**Figura 52.** Consulta utilizando dois conceitos com CONTAINSTABLE

```
SELECT *
FROM Resposta AS FT_TBL
  INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
    'FORMSOF(THESAURUS, "receita de vendas")
  OR FORMSOF(THESAURUS, "expectativa de lucro"')) AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
```

Fonte: Desenvolvido pelo autor

A Figura 53 exibe o resultado da consulta utilizando OR. É possível observar que o comentário 3 é retornado ao utilizar o operador OR, diferente do resultado utilizando o operador AND, que retorna os comentários 1, 2 e 4.

**Figura 53.** Resultados obtidos na consulta da Figura 52

KEY	RANK
2	13
3	6
4	6
1	6

Fonte: Desenvolvido pelo autor

## 4.2 Resultados obtidos

As duas consultas propostas na seção 4.1 se mostraram úteis para a recuperação de textos por meio da similaridade de conceitos. Ambas as consultas possibilitam a busca por sinônimos. Entretanto, a Consulta 2 não considera a flexão dos termos dos conceitos. Isso é possível observar, por exemplo, nos resultados das consultas pelo conceito “receita de vendas”, onde a relevância é diferente quando “receitas de vendas” está presente no texto. Diante dessas considerações, a Consulta 1, que utiliza a função FREETEXTTABLE, se mostra mais adequada, dado que considera sinônimos e flexão dos termos consultados.

## 5. CONSIDERAÇÕES FINAIS

Este trabalho apresentou os recursos oferecidos pelo SGBD SQL Server para dados do tipo *Full Text* e como podem ser utilizados para a recuperação de textos por meio da similaridade de conceitos.

Foi realizado um estudo de caso, utilizando um banco de dados com os comentários de um blog. A partir dos recursos apresentados foi possível definir as consultas mais adequadas para a recuperação de textos usando conceitos.

Os recursos de pesquisa semântica oferecidos pelo SGBD SQL Server não permitem a busca por frases, o que inviabilizou a sua utilização no contexto deste trabalho. Por outro lado, os recursos oferecidos para dados *Full Text* se mostraram apropriados.

Para trabalhos futuros sugere-se o desenvolvimento de um software, de modo a facilitar a execução das consultas por meio de uma interface gráfica.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

EISENBERG, Andrew; MELTON, Jim. **SQL:1999, formerly known as SQL3**. 1999. Disponível em: < <https://www.cl.cam.ac.uk/teaching/2003/Databases/sql1999.pdf> >. Acesso em: 09 de Agosto de 2021.

ELMASRI, Ramez; NAVATHE, Shamkant B., **Fundamentals of Database Systems**. 7 ed., Pearson, 2016.

MELTON, Jim; EISENBERG, Andrew., **SQL Multimedia and Application Packages (SQL/MM)**, SIGMOD Record, v. 30, p. 97-102, 2001.

PAIVA, Irlisene S., **Estudo de Pesquisa Semântica em Sistemas de Gerenciamento de Banco de Dados e sua aplicabilidade na avaliação de questões discursivas**. Trabalho de Conclusão de Curso, Faculdade de Tecnologia da Unicamp, Limeira, 2020.

ASHRAF, Harron. **How to Use Semantic Search in SQL Server**. Disponível em: <<https://codingsight.com/how-to-use-semantic-search-in-sql-server/>>. Acesso em: 10 de Agosto de 2021.

MICROSOFT. **Pesquisa de Texto Completo**. 2020. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/search/full-text-search?view=sql-server-ver15>>. Acesso em 19 de Agosto de 2021.

MICROSOFT. **Query with Full-Text Search**. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/search/query-with-full-text-search?view=sql-server-ver15> >. Acesso em 19 de Agosto de 2021.

SUSANTO, Azhar; BONG, Meiryani. **Database Management System**. 2019. Disponível em: < <http://www.ijstr.org/final-print/june2019/Database-Management-System.pdf> >.

Acesso em 27 de Agosto de 2021.

ORACLE. **Database SQL Reference**. 2021. Disponível em: [https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/intro001.htm#i1712](https://docs.oracle.com/cd/B19306_01/server.102/b14200/intro001.htm#i1712) >. Acesso em 05 de Setembro de 2021.

LIMA, Allyn. **Padrão SQL e sua Evolução**. Disponível em: < <https://www.ic.unicamp.br/~geovane/mo410-091/Ch05-PadraoSQL-art.pdf> > Acesso em 05 de Setembro de 2021.

MELTON, Jim. **Advanced: SQL:1999**. United States of America: Elsevier Science, 2003.

MANNING, Christopher; RAGHAVAN; SCHUTZE, Hinrich. **An Introduction to Information Retrieval**. England. Cambridge University Press, 2009.

CODD, Edgar. **A Relational Model of Data for Large Shared Data Banks**. Communications of the ACM, 1970.

ASHRAF, Harron. **How to Use Semantic Search in SQL Server**. 2020. Disponível em: <<https://codingsight.com/how-to-use-semantic-search-in-sql-server/>>. Acesso em: 1 de Agosto de 2021.

MICROSOFT. **Instalar e configurar a pesquisa semântica**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/search/install-and-configure-semantic-search?view=sql-server-ver15> >. Acesso em: 1 de Agosto de 2021.

MICROSOFT. **Criar e gerenciar catálogos de texto completo**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/search/create-and-manage-full-text-catalogs?view=sql-server-ver15>>. Acesso em: 1 de Agosto de 2021.

MICROSOFT. **Criar e gerenciar índices de texto completo**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/search/create-and-manage-full-text-indexes?view=sql-server-ver15>>. Acesso em: 1 de Agosto de 2021.

MICROSOFT. **Habilitar a pesquisa semântica em tabelas e colunas**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/search/enable-semantic-search-on-tables-and-columns?view=sql-server-ver15>>. Acesso em: 22 de Agosto de 2021.

MICROSOFT. **sys.fulltext\_languages**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/system-catalog-views/sys-fulltext-languages-transact-sql?view=sql-server-ver15>>. Acesso em: 22 de Agosto de 2021.

Fronc, Arkadiusz. **Full-Text Search with PDF documents in SQL Server 2014**. 2015. Disponível em: < <https://www.youtube.com/watch?v=JjO79utFDUE>>. Acesso em: 20 de Março de 2022.

MICROSOFT. **Localizar frases chave em documentos com pesquisa semântica**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/search/find-key-phrases-in-documents-with-semantic-search?view=sql-server-ver15>>. Acesso em 23 de Março de 2022.

MICROSOFT. **semantickeyphrasetable**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/system-functions/semantickeyphrasetable-transact-sql?view=sql-server-ver15>>. Acesso em 05 de Maio de 2022.

MICROSOFT. **Limite resultados de pesquisa com RANK**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/search/limit-search-results-with-rank?view=sql-server-ver155>>. Acesso em 05 de Abril de 2022.

MICROSOFT. **FREETEXTTABLE**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/system-functions/freetexttable-transact-sql?view=sql-server-ver15>>. Acesso em 05 de Abril de 2022.

MICROSOFT. **semanticsimilaritytable**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/system-functions/semanticsimilaritytable-transact-sql?view=sql-server-ver15>>. Acesso em 05 de Abril de 2022.

MICROSOFT. **ISABOUT Term**. 2021. Disponível em: < <https://docs.microsoft.com/en-us/windows/win32/search/-search-sql-isabout>>. Acesso em 11 de Abril de 2022.

MICROSOFT. **CONTAINS**. 2021. Disponível em: < <https://docs.microsoft.com/pt-br/sql/t-sql/queries/contains-transact-sql?view=sql-server-ver15>>. Acesso em 13 de Abril de 2022.

MICROSOFT. **FORMSOF Term**. 2021. Disponível em: < <https://docs.microsoft.com/en-us/windows/win32/search/-search-sql-formsof>>. Acesso em 13 de Abril de 2022.

MICROSOFT. **Configurar e gerenciar arquivos de dicionário de sinônimos para Pesquisa de texto completo**. 2022. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/search/configure-and-manage-thesaurus-files-for-full-text-search?view=sql-server-ver15>>. Acesso em 13 de Abril de 2022.

MICROSOFT. **Usar a pesquisa de texto completo com colunas XML**. 2022. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/xml/use-full-text-search-with-xml-columns?view=sql-server-ver15>>. Acesso em 13 de Abril de 2022.

MICROSOFT. **Procurar palavras perto de outra palavra com NEAR**. 2022. Disponível em: < <https://docs.microsoft.com/pt-br/sql/relational-databases/search/search-for-words-close-to-another-word-with-near?view=sql-server-ver16>>. Acesso em 13 de Abril de 2022

## APÊNDICE A – Instalação e configuração de recursos *Full Text*

A instalação do SQL Server e os recursos para *Full Text* foram realizadas seguindo o tutorial presente no site [codingsight.com](http://codingsight.com).

A versão mais atual que possibilitou as buscas semânticas foi SQL Server 2016, as versões 2017 e 2019 exibem erros ao realizar a instalação dos recursos para *Full Text*. Importante lembrar que a instalação dos recursos para *Full Text* deve ser realizada juntamente com a instalação do SQL Server.

```
use master
```

```
--Verificar se o recurso FullText Search está instalado
```

```
SELECT SERVERPROPERTY('IsFullTextInstalled') as  
[Full-Text-Search-and-Semantic-Search-Installed];
```

```
-- Criar o banco de dados
```

```
CREATE DATABASE BD_Resposta
```

```
USE BD_Resposta
```

```
-- Criação de tabela de documentos
```

```
CREATE TABLE Resposta(  
    Id INT IDENTITY(1,1) NOT NULL,  
    Conteudo VARCHAR(1000) NOT NULL,  
    CONSTRAINT PK_BD_Resposta PRIMARY KEY CLUSTERED(Id)  
);
```

```
-- Inserção dos dados na tabela
```

```
INSERT INTO Resposta(Conteudo)
```

```
VALUES ('Olá Luiz! Boa questão. Nesse âmbito, a expectativa de lucro, por parte dos stakeholders/sócios afeta sim o preço do produto, pois, para que exista o lucro, a soma das despesas e dos custos não deve ser maior que o valor da receita de vendas. Esse retorno é importante para os stakeholders pois demonstra que o produto em que ele investiu teve seu market share. A importância dos stakeholders, assim como os colegas disseram acima, varia com a situação, sobre quais eles podem fazer diferentes papéis.')
```

INSERT INTO Resposta(Conteudo)

VALUES ('Olá Pedro, em relação a sua dúvida do CMAPS, o sentido do fluxo de capital ser negativo é que existe fluxo de capital negativo e temos que considera-lo. No fluxo de capital existem dois fluxos: o de entrada e saída. Fluxo de entrada vem das receitas de vendas e o de saída dos custos e despesas. Diante disso a soma dos custos e despesas não pode ultrapassar o valor da Receita de vendas, pois é necessário que a empresa tenha LUCRO, para contribuir com impostos. Para entregarmos valor precisamos ser atrativos, e para isso tem que ter uma expectativa de lucro real, ou seja, FLUXO DE CAPITAL POSITIVO(receita de vendas maior que Despesas + custos.)

INSERT INTO Resposta(Conteudo)

VALUES ('Bom dia, eu fiquei com uma dúvida em relação a possíveis sócios da empresa que auxiliariam com o Capital Social. O preço do produto deve ser comparado com o preço do concorrente e é composto por lucro, custos, despesas e impostos porém ainda devemos nos atentar para a expectativa de lucro dos sócios e ainda gerar valor para empresa. Minha dúvida é se a expectativa de lucro dos sócios afeta diretamente o preço do produto, pois uma vez que investiram na empresa esperam um retorno válido, o que pode acabar afetando a atratividade de mercado do nosso produto e diminuir nosso diferencial competitivo nesse quesito. Sendo assim se a empresa tiver a possibilidade de iniciar sem a ajuda de um sócio seria melhor? ou é melhor tentar balancear a expectativa dos sócios com as necessidades da empresa, mesmo se a empresa tiver condições para iniciar sem a necessidade de uma sociedade?')

INSERT INTO Resposta(Conteudo)

VALUES ('Olá pessoal, achei uma boa ideia o ponto levantado pelo Victor e comentado pelo Arthur. Se a Hexágono buscar por membros que realmente são bons em publicidade, isso levará um aumento da expectativa de lucro real, o que, por sua vez, aumenta a atratividade do mercado. Isso ocorre porque o fluxo de capital (Receita de vendas > Despesas + Custos) será positivo e superior aos outros tipos de investimentos, uma vez que estaríamos diminuindo os custos. Ressalta-se que isso só será possível se os membros contratados forem empenhados e saberem trabalhar com publicidade, caso contrário voltamos ao estágio inicial. Sobre a proposta de dar bolsas para estes membros, citada pelo Arthur, também concordo que seria um ótimo cenário para ambas as partes. Alguém tem alguma outra ideia para contratar estes membros?')

-- Criar a database semanticfdb

CREATE DATABASE semanticfdbResposta ON

(FILENAME = 'C:\Program Files\Microsoft Semantic Language Database\semanticfdb.mdf'),

(FILENAME = 'C:\Program Files\Microsoft Semantic Language Database\semanticfdb\_log.ldf')

FOR ATTACH;

-- Local da pasta de log semantic

C:\Program Files\Microsoft SQL Server\MSSQL13.SQLEXPRESS\MSSQL\DATA

Possível erro de permissão:

```
-- Instalar SemanticLanguageDatabase.msi -- https://codingsight.com/how-to-use-semantic-search-in-sql-server
-- Criar a database semanticfdb
CREATE DATABASE semanticfdbBDResposta ON
(FILENAME = 'C:\Program Files\Microsoft Semantic Language Database\semanticfdb.mdf'),
(FILENAME = 'C:\Program Files\Microsoft Semantic Language Database\semanticfdb_log.ldf')
FOR ATTACH;
```

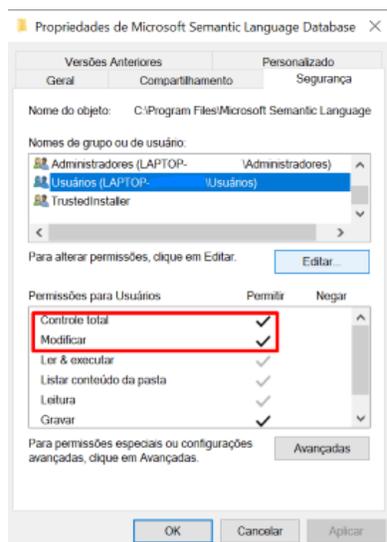
100 %

Messages

Msg 3415, Level 16, State 2, Line 77  
O banco de dados 'semanticfdbBDResposta' não pode ser atualizado porque é somente leitura, tem arquivos somente leitura

Completion time: 2022-04-17T21:15:49.4103077-03:00

Para contornar o erro é necessário alterar as permissões de acesso/modificação a pasta “Microsoft Semantic Language Database”, basta ir nas propriedades -> segurança -> editar, clique no seu usuário e selecione as opções destacadas na imagem caso:



-- Registrar a database semanticfdb

EXEC sp\_fulltext\_semantic\_register\_language\_statistics\_db @dbname = N'semanticfdbBDResposta';

-- Criar catalogo fulltext

CREATE FULLTEXT CATALOG catalogofulltext;

---- Criação de índice Full Text

```
CREATE FULLTEXT INDEX ON Resposta (  
    Conteudo    -- coluna que será indexada  
    Language 1046 STATISTICAL_SEMANTICS -- 1046 - LCID Brasil - portugues  
)
```

```
KEY INDEX PK_BD_Resposta ON catalogofulltext --Unique index
```

```
WITH CHANGE_TRACKING AUTO    -- Population type;
```

```
-- Carregar os resources
```

```
-- Após a carga dos resources reiniciar o computador
```

```
EXEC sys.sp_fulltext_service 'load_os_resources', 1;
```

```
--Listando todos os componentes de texto completo do sistema
```

```
EXEC sys.sp_help_fulltext_system_components 'filter';
```

Para configurar o dicionário de sinônimos, deve se localizar o arquivo do idioma desejado, o caminho varia de acordo com a versão do SQL Server utilizada.

C:\Program Files\Microsoft SQL Server\MSSQL13.SQLEXPRESS\MSSQL\FTData

O arquivo referente ao idioma português – Brasil é o “tsptb”, o mesmo já possui a formatação básica do dicionário XML. O arquivo foi editado de acordo com os termos relevantes para o estudo de caso.

```
<XML ID="Microsoft Search Thesaurus">
```

```
<thesaurus xmlns="x-schema:tsSchema.xml">  
    <diacritics_sensitive>0</diacritics_sensitive>  
    <expansion>  
        <sub>tributos</sub>  
        <sub>impostos</sub>  
        <sub>taxas</sub>  
    </expansion>  
    <expansion>  
        <sub>atratividade</sub>  
        <sub>atrativos</sub>  
    </expansion>  
    <expansion>
```

```

    <sub>investimento</sub>
    <sub>capital</sub>
</expansion>
    <expansion>
    <sub>preço</sub>
    <sub>custos</sub>
        <sub>valor</sub>
        <sub>despesas</sub>
    </expansion>
    <expansion>
    <sub>vendas</sub>
    <sub>lucro</sub>
        <sub>receita</sub>
        <sub>produto</sub>
    </expansion>
    <expansion>
    <sub>sócio</sub>
    <sub>membros</sub>
</expansion>
    <expansion>
    <sub>empresa</sub>
    <sub>sociedade</sub>
</expansion>
</thesaurus>
</XML>

```

-- Carregar um arquivo de dicionário de sinônimos para o idioma desejado.

-- Caso o arquivo seja editado novamente, o comando deve ser executado novamente.

```
EXEC sys.sp_fulltext_load_thesaurus_file 1046;
```

--Gerenciamento de índices Full Text

-- Retornar informações sobre as populações de índice de texto completo que estão em andamento.

```
SELECT * FROM sys.dm_fts_index_population WHERE table_id =
OBJECT_ID('Resposta')
```

-- Retornar uma linha de informações de status sobre a população do índice de similaridade de documento para cada índice de similaridade em cada tabela que tem um índice semântico associado.

```
SELECT * FROM sys.dm_fts_semantic_similarity_population WHERE table_id =  
OBJECT_ID('Resposta')
```

-- Retornar informações de tamanho e fragmentação para os dados e índices da tabela ou exibição especificada no SQL Server

```
SELECT * FROM sys.dm_db_fts_index_physical_stats WHERE object_id =  
OBJECT_ID('Resposta')
```

--listar palavras indexadas

```
SELECT * FROM sys.dm_fts_index_keywords( DB_ID('BD_Resposta'),  
OBJECT_ID('Resposta') )
```

```
ORDER BY document_count DESC
```

## APÊNDICE B – Configuração para consultas a PDF e HTML

Para consultas em documentos HTML realizar os seguintes passos.

--Verificar se o filtro HTML está instalado

```
SELECT * FROM sys.fulltext_document_types WHERE document_type = '.html'
```

--Criar tabela que irá armazenar os documentos

```
CREATE TABLE Posts(  
    [Id] INT NOT NULL,  
    [Title] NVARCHAR(MAX) NOT NULL,  
    [Content] NVARCHAR(MAX) NOT NULL,  
    CONSTRAINT Pk_Posts PRIMARY KEY (Id)  
)
```

--Converter as colunas para varbinary(max)

```
ALTER TABLE Posts
```

```
DROP COLUMN [Title]
```

```
ALTER TABLE Posts
```

```
DROP COLUMN [Content]
```

```
ALTER TABLE Posts
```

```
ADD [Title] VARBINARY(MAX) NOT NULL
```

```
ALTER TABLE Posts
```

```
ADD [Content] VARBINARY(MAX) NOT NULL
```

--Adicionar uma coluna para especificar o filtro HTML

```
ALTER TABLE Posts
```

```
Add FileExtension As '.html'
```

--Inserir dados na tabela

```
INSERT INTO Posts
```

```
VALUES (1, CONVERT(VARBINARY(MAX), 'Some title'),  
CONVERT(VARBINARY(MAX), '<p><strong>f</strong>oo</p>'))
```

--Criar o índice Full-Text para as colunas "Title" e "Content"

```
CREATE FULLTEXT CATALOG [posts_catalog]
```

```
CREATE FULLTEXT INDEX ON Posts
```

```
(
```

```
    [Title] TYPE COLUMN FileExtension, [Content] TYPE COLUMN FileExtension
```

```
)
```

```
    KEY INDEX Pk_Posts ON posts_catalog
```

--Consulta pelo termo foo, flexionado ou não

```
SELECT
```

```
    Id,
```

```
    CONVERT(VARCHAR(MAX), [Title]) AS [Title],
```

```
    CONVERT(VARCHAR(MAX), [Content]) AS [Content]
```

```
FROM Posts
```

```
WHERE FREETEXT (*,'foo')
```

Para consultas em documentos PDF é necessário instalar o plug-in para PDF Ifilter PDF, adicionar o caminho da pasta bin nas variáveis de ambiente do sistema operacional, e executar os seguintes comandos (FRONC ,2015).

--Carregar recursos

```
EXEC sp_fulltext_service @action='load_os_resources', @value=1;
```

--Desativar verificação de login para evitar problemas ao carregar filtros

```
EXEC sp_fulltext_service @action='verify_signature', @value=0;
```

--Reiniciar o computador para aplicar as mudanças

--Verificar se o tipo PDF está instalado

```
SELECT document_type, path
```

```
FROM sys.fulltext_documents_types
```

```
WHERE document_type = '.pdf'
```

--Criar tabela para armazenar os arquivos PDF

```

CREATE TABLE dbo.articles(
    id INT IDENTITY(1,1) NOT NULL,
    content VARBINARY(MAX),
    extension VARCHAR(5),
    title NVARCHAR(255)
);

ALTER TABLE dbo.articles ADD CONSTRAINT PK_articles PRIMARY KEY
CLUSTERED(
    id
);

--Inserir documentos na tabela

INSERT INTO dbo.articles (content,extension,title)
SELECT bulkcolumn, '.pdf', 'Generating XML from SQL part 1'
FROM OPENROWSET(BULK 'c:\temp\SQL - Generating XML from SQL part 1.pdf',
SINGLE_BLOB) AS t;

INSERT INTO dbo.articles (content,extension,title)
SELECT bulkcolumn, '.pdf', 'Generating XML from SQL part 2'
FROM OPENROWSET(BULK 'c:\temp\SQL - Generating XML from SQL part 2',
SINGLE_BLOB) AS t;

--Verificar se os dados estão na tabela, a coluna contente estará em binário
SELECT *
FROM articles

--Criar catálogo Full Text
CREATE FULLTEXT CATALOG SqlArticles;

--Criar índice Full Text, código 1033 corresponde a língua inglesa
CREATE FULLTEXT INDEX ON dbo.articles(
    content TYPE COLUMN extension LANGUAGE 1033
)
KEY INDEX PK_articles ON SqlArticles;

```

--Pode-se verificar se a execução ocorreu corretamente na pasta de LOG do Microsoft SQL Server

--Pode-se verificar também através de uma consulta por Keywords

```
SELECT *
```

```
FROM sys.dm_fts_index_keywords(DB_ID('MyDatabase'),OBJECT_ID('articles'))
```

--Realizar consultas Full Text

```
SELECT *
```

```
FROM dbo.articles
```

```
WHERE CONTAINS (content,'xml');
```

## APÊNDICE C – Consultas realizadas no estudo de caso

```
use BD_Resposta
```

```
-- TOPICO 1: ATRATIVIDADE DO MERCADO
```

```
--C1
```

```
--FREETEXTTABLE
```

```
SELECT *
```

```
FROM Resposta AS FT_TBL
```

```
    INNER JOIN FREETEXTTABLE(Resposta, *, 'atratividade do mercado') AS  
KEY_TBL
```

```
    ON FT_TBL.Id = KEY_TBL.[KEY]
```

```
    ORDER BY KEY_TBL.RANK DESC
```

```
--C2
```

```
--CONTAINSTABLE, FORMSOF
```

```
SELECT *
```

```
FROM Resposta AS FT_TBL
```

```
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
```

```
'FORMSOF(THESAURUS, "atratividade de mercado") ') AS KEY_TBL
```

```
    ON FT_TBL.Id = KEY_TBL.[KEY]
```

```
    ORDER BY KEY_TBL.RANK DESC
```

```
-- TOPICO 2: CAPITAL ECONÔMICO
```

```
--FREETEXTTABLE
```

```
SELECT *
```

```
FROM Resposta AS FT_TBL
```

```
    INNER JOIN FREETEXTTABLE(Resposta, *, 'capital econômico') AS KEY_TBL
```

```
    ON FT_TBL.Id = KEY_TBL.[KEY]
```

```
    ORDER BY KEY_TBL.RANK DESC
```

```
--CONTAINSTABLE, FORMSOF
```

```
SELECT *
```

```
FROM Resposta AS FT_TBL
```

```
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
```

```

        'FORMSOF(THESAURUS, "capital econômico") ') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]
        ORDER BY KEY_TBL.RANK DESC
-- TOPICO 3: CAPITAL INTELECTUAL
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
        INNER JOIN FREETEXTTABLE(Resposta, *, 'capital intelectual') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]
        ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
        INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "capital intelectual") ') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]
        ORDER BY KEY_TBL.RANK DESC
-- TOPICO 4: CAPITAL SOCIAL
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
        INNER JOIN FREETEXTTABLE(Resposta, *, 'capital social') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]
        ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
        INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "capital social") ') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]

```

```

ORDER BY KEY_TBL.RANK DESC
-- TOPICO 5: CUSTO DE PRODUÇÃO
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN FREETEXTTABLE(Resposta, *, 'custo de produção') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
'FORMSOF(THESAURUS, "custo de produção" )') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
-- TOPICO 6: DESPESAS
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN FREETEXTTABLE(Resposta, *, 'despesas') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
'FORMSOF(THESAURUS, "despesas" )') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
-- TOPICO 7: EXPECTATIVA DE LUCRO

```

```
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN FREETEXTTABLE(Resposta, *, 'expectativa de lucro') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
```

```
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
    'FORMSOF(THESAURUS, "expectativa de lucro") ') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
```

-- TOPICO 8: FLUXO DE CAPITAL

```
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN FREETEXTTABLE(Resposta, *, 'fluxo de capital') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
```

```
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
    'FORMSOF(THESAURUS, "fluxo de capital") ') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
```

```

-- TOPICO 9: INOVAÇÃO
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN FREETEXTTABLE(Resposta, *, 'inovação') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
    'FORMSOF(THESAURUS, "inovação") ') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
-- TOPICO 10: INVESTIMENTO
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN FREETEXTTABLE(Resposta, *, 'investimento') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
    'FORMSOF(THESAURUS, "investimento") ') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
-- TOPICO 11: LUCRO
--FREETEXTTABLE

```

```

SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN FREETEXTTABLE(Resposta, *, 'lucro') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
    'FORMSOF(THESAURUS, "lucro") ') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
-- TOPICO 12: PREÇO DO PRODUTO
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN FREETEXTTABLE(Resposta, *, 'preço do produto') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
    'FORMSOF(THESAURUS, "preço do produto") ') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
-- TOPICO 13: RECEITA DE VENDAS
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL

```

```

INNER JOIN FREETEXTTABLE(Resposta, *, 'receita de vendas') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
'FORMSOF(THESAURUS, "receita de vendas") ') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
-- TOPICO 14: STAKEHOLDERS
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN FREETEXTTABLE(Resposta, *, 'stakeholders') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
'FORMSOF(THESAURUS, "stakeholders") ') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]
ORDER BY KEY_TBL.RANK DESC
-- TOPICO 15: SÓCIO
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
INNER JOIN FREETEXTTABLE(Resposta, *, 'sócio') AS KEY_TBL
ON FT_TBL.Id = KEY_TBL.[KEY]

```

```

        ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
        INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "sócio" ) ') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]
        ORDER BY KEY_TBL.RANK DESC
-- TOPICO 16: TRIBUTOS
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
        INNER JOIN FREETEXTTABLE(Resposta, *, 'tributos') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]
        ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF
SELECT *
FROM Resposta AS FT_TBL
        INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "tributos" ) ') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]
        ORDER BY KEY_TBL.RANK DESC
-- TOPICO 17: VALOR PARA A EMPRESA
--FREETEXTTABLE
SELECT *
FROM Resposta AS FT_TBL
        INNER JOIN FREETEXTTABLE(Resposta, *, 'valor para a empresa') AS KEY_TBL
        ON FT_TBL.Id = KEY_TBL.[KEY]
        ORDER BY KEY_TBL.RANK DESC
--CONTAINSTABLE, FORMSOF

```

```
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "valor para empresa") ') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
```

--Consulta utilizando dois conceitos com AND

```
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "receita de vendas")
        AND FORMSOF(THESAURUS, "expectativa de lucro")') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
```

--Consulta utilizando dois conceitos com OR

```
SELECT *
FROM Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(Resposta, Conteudo,
        'FORMSOF(THESAURUS, "receita de vendas")
        OR FORMSOF(THESAURUS, "expectativa de lucro")') AS KEY_TBL
    ON FT_TBL.Id = KEY_TBL.[KEY]
    ORDER BY KEY_TBL.RANK DESC
```