

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE TECNOLOGIA

**ESTUDO DE PESQUISA SEMÂNTICA EM SISTEMAS DE
GERENCIAMENTO DE BANCO DE DADOS E SUA APLICABILIDADE
NA AVALIAÇÃO DE QUESTÕES DISCURSIVAS**

IRISLENE SILVEIRA DE PAIVA

LIMEIRA, 2020

IRISLENE SILVEIRA DE PAIVA

**ESTUDO DE PESQUISA SEMÂNTICA EM SISTEMAS DE
GERENCIAMENTO DE BANCO DE DADOS E SUA APLICABILIDADE
NA AVALIAÇÃO DE QUESTÕES DISCURSIVAS**

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
2020

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, que me capacitou para realizar este trabalho e aos meus familiares e amigos que me apoiaram nessa fase da minha vida. Quero agradecer, especialmente, a minha irmã Neyrielen Silveira de Paiva e gostaria de mencionar alguns nomes de amigos que foram fundamentais nesse processo, Alfredo Albélis Batista Filho, Bruno do Valle Saes Adegas, Mateus Soraggi Battagin, Miguel Mechi Naves Rocha, e Pedro Henrique de Souza Pozelli. Por fim, agradeço, a Profa. Dra. Gisele Busichia Baioco, por ter sido a minha orientadora e sempre ter me ajudado com muita paciência na execução desse estudo, e também aos professores que participaram da minha banca examinadora: Prof. Dr. Antônio Carlos Zambon e Prof. Dr. Luiz Camolesi Júnior.

RESUMO

Pensando em otimizar o tempo de trabalho dos avaliadores na correção de questões discursivas, este trabalho analisou as funcionalidades oferecidas pelos Sistemas de Gerenciamento de Banco de Dados Oracle, SQL Server e DB2 em relação ao tipo de dado *Full Text*. Após essa análise, o SQL Server foi escolhido, e suas funcionalidades que envolvem a Pesquisa Semântica foram aplicadas em um estudo de caso relacionado à avaliação de questões discursivas, por meio de um banco de dados composto por respostas com peculiaridades bem definidas (certa, meio certa, incompleta e incorreta), desenvolvido pela autora. Espera-se que com os resultados obtidos, os recursos de Pesquisa Semântica estudados e apresentados neste trabalho possam ser utilizados futuramente como uma ferramenta auxiliadora para a avaliação semiautomática desse tipo de questão.

Palavras-chave: SQL Server, Oracle, DB2, pesquisa semântica, avaliação semiautomática, questão discursiva, *Full Text*.

STUDY OF SEMANTIC SEARCH IN DATABASE MANAGEMENT SYSTEMS AND THEIR APPLICABILITY IN THE EVALUATION OF DISCURSIVE QUESTIONS

ABSTRACT

Thinking in optimizing the evaluators' working time in correcting discursive questions, this study analyzed the functionalities offered by databases Oracle, SQL Server and DB2 in relation to the Full Text data type. After this analysis, SQL Server was chosen, and the functionalities that involved Semantic Search were applied in a case study related to the evaluation of discursive questions, through a database composed of answers with well-defined peculiarities (right, not fully right, incomplete and incorrect), developed by the author. It is expected that with the results obtained, the Semantic Research resources studied and presented in this study can be used in the future as an auxiliary tool for the semi-automatic evaluation of this type of question.

Keywords: SQL Server, Oracle, DB2, Semantic Search, evaluation semi-automatic, discursive questions, Full Text.

SUMÁRIO

1 INTRODUÇÃO	7
2 A LINGUAGEM SQL	9
2.1 SQL: 1999: a SQL com características de orientação a objetos	9
2.2 SQL/MM e o tipo de dado <i>Full Text</i>	10
3 ESTUDO DOS SGBD	16
3.1 Análise dos principais SGBD Objeto-Relacionais em relação aos recursos propostos no padrão SQL/MM	16
3.2 Estudo mais aprofundado dos SGBD Objeto-Relacionais Oracle, SQL Server e DB2	19
3.2.1 DB2	19
3.2.2 Oracle	21
3.2.3 SQL Server	26
3.3 Resultados e conclusões	29
4 ESTUDO DE CASO: COMO A PESQUISA SEMÂNTICA PODE AJUDAR NA AVALIAÇÃO DE QUESTÕES DISCURSIVAS?	32
4.1 Primeira possibilidade de análise: o avaliador informa as palavras relevantes	34
4.2 Segunda possibilidade de análise: respostas que o SQL Server compara automaticamente sem a necessidade de especificar palavras-chave	41
5 CONSIDERAÇÕES FINAIS	51
6 REFERÊNCIAS BIBLIOGRÁFICAS	52
APÊNDICE	58

1. INTRODUÇÃO

A partir do armazenamento de dados do tipo *Full Text* (texto completo), como por exemplo: livros, dissertações, revistas e jornais, entre outros tipos de documentos textuais, em Banco de Dados, surge a necessidade da utilização de mecanismos de busca que permitam uma análise/consulta de forma rápida e eficiente a esse tipo de dado. Para isso, a Pesquisa Semântica, ou também conhecida como Pesquisa com Significado, é um termo que ganhou relevância nos últimos tempos, e se apresenta como uma maneira de resolver e atender às necessidades de busca atuais nos Sistemas de Gerenciamento de Banco de Dados (SGBD). A pesquisa semântica, vai além do estudo do significado das palavras, e surge com o intuito de melhorar a precisão das pesquisas a dados textuais longos, proporcionando, por exemplo, a possibilidade do entendimento do contexto da consulta e a relação entre as palavras pesquisadas (BERASATEGI, 2019).

Alguns SGBD relacionais possuem recursos padronizados para a implementação das consultas a dados *Full Text*, podendo ser denominados SGBD objeto-relacionais (Eisenberg e Melton, 1998). Desse modo, o objetivo deste trabalho foi analisar comparativamente SGBD objeto-relacionais, verificando os recursos oferecidos por estes e, a partir disso, foram selecionados os que apresentaram mecanismos de consulta para dados *Full Text*. Em seguida, foi escolhido um SGBD, sobre o qual foi realizado um estudo mais detalhado, aplicando a pesquisa semântica em um estudo de caso e avaliando a sua capacidade para a avaliação de questões discursivas.

A monografia foi estruturada nos seguintes capítulos, além desta introdução:

Capítulo 2 - A linguagem SQL: Contextualização dos padrões SQL:1999 e SQL/MM. Apresentação das funcionalidades definidas pelo padrão SQL/MM em relação ao tipo de dado *Full Text*.

Capítulo 3 – Estudo dos SGBD: Análise dos principais SGBD objeto-relacionais em relação aos recursos propostos pelo padrão SQL/MM e estudo dos SGBD DB2, Oracle e SQL Server. Definição do SGBD para aplicação no estudo de caso.

Capítulo 4 – Estudo de Caso: Como a pesquisa semântica pode ajudar na avaliação de questões discursivas? Aplicação das funcionalidades dos SGBD

relacionadas à pesquisa semântica que podem ser úteis na avaliação de questões discursivas.

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

2. A LINGUAGEM SQL

Com o intuito de compreender melhor os dados multimídia e como são tratados pelos SGBDs Objeto-Relacionais, nesse capítulo, foram abordadas bibliografias sobre as versões SQL:1999 e SQL/MM, com ênfase no padrão SQL/MM, que propõe diversas funcionalidades em relação ao tipo de dado *Full Text*.

2. 1. SQL:1999: a SQL com características de orientação a objetos

A linguagem SQL passou por diversas versões no decorrer de sua história, tornando-se a linguagem comercial de banco de dados mais utilizada nos dias de hoje. Organizações como ANSI (*American National Standards Institute*) e ISO (*International Organization for Standardization*) foram responsáveis pela sua padronização, desenvolvendo atualizações desde 1993. Uma dessas atualizações deu origem ao padrão SQL:1999, também conhecido como SQL3 (LIMA, 2004).

O padrão SQL:1999 foi uma versão da linguagem SQL desenvolvida com o objetivo de ser muito mais que apenas um aprimoramento da versão anterior (SQL segunda geração ou SQL-92), e ficou conhecido popularmente como “SQL orientada a objetos”, tornando-se a base para os SGBD objeto-relacionais.

Recursos foram adicionados ao SQL: 1999, como por exemplo, os tipos de dados *Large Object* (LOB), *Array*, *Row* e *Boolean*, consultas recursivas, gatilhos e características de orientação a objetos. Dentro do conceito orientação a objetos, foi criado o *User Defined Types* – UDT, tipos de dados definidos pelo usuário, e o *User Defined Functions* – UDF, funções que podem ser definidas pelo usuário.

O UDF configura-se como uma função que pode ser escrita em SQL ou em linguagens de programação (ex.: Java), capaz de permitir a incorporação de um código escrito pelo usuário que auxilia os processos de armazenamento e a recuperação de informação no banco de dados.

Já o UDT é o recurso mais fundamental no SQL:1999; através dele é possível a padronização do desenvolvimento de objetos no Banco de Dados. Com o suporte à herança simples e tendo todos os seus comportamentos fornecidos por métodos, funções e procedimentos, o UDT possibilita a criação de tipos (subtipos), que além de possuírem seus próprios atributos e rotinas, apresentam todos eles associados ao subtipo mais especializado (supertipo). Ou seja, uma vez que temos uma hierarquia

onde “funcionário” é um subtipo que herda do supertipo “pessoa comum”, ele possuirá todos os atributos contidos em “pessoa comum” e seus próprios atributos, por exemplo, “id_funcionario” e “salário”.

2. 2. SQL/MM e o tipo de dado *Full Text*

O padrão SQL Multimídia ou também chamado SQL/MM, foi um padrão baseado na linguagem SQL:1999 e seus UDT. A partir dele foi possível obter o tratamento de dados multimídia, como por exemplo: *Full Text*, para o processamento de texto, *Spatial* para o processamento de coordenadas, e o *Still Image* para o processamento de imagem. Entretanto, neste trabalho abordaremos apenas o tipo de dado *Full Text*.

O tipo de dado *Full Text* (texto completo) difere das tradicionais cadeias de caracteres em relação ao seu tamanho e por permitir a aplicação de operações específicas, as quais discutiremos mais adiante. Outra diferença se refere à indexação. Enquanto cadeias de caracteres são indexadas pelo seu valor inteiro, 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 até mesmo em documentos diferentes (MELTON e EISENBERG, 2001). É importante salientar que apesar dos índices serem de extrema importância, o padrão SQL/MM não usa e nem oferece suporte a eles. (MELTON, 2003).

Como foi dito anteriormente, o padrão SQL/MM estabelece diversos UDT. Entre eles, temos o UDT *Full Text*, que possibilita o uso de dados de texto completo por meio de métodos para a realização de operações nesse tipo de dado. Algumas dessas operações podem ser bastante complexas, permitindo, por exemplo, 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.

Segundo Melton e Eisenberg (2001), apesar de existirem idiomas com menor propensão à identificação de componente de texto (por exemplo, o idioma japonês não possui espaço em branco para separar as palavras), o SQL/MM ainda foi reconhecido como o melhor para o suporte de idiomas.

A Figura 1 apresenta um exemplo de criação de uma tabela denominada “information” que contém as colunas “docno” do tipo inteiro, para armazenar um número identificador para cada documento, e “document” do tipo *Full Text*, para armazenar os textos dos documentos.

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

```
CREATE TABLE information (  
  docno INTEGER;  
  document FULLTEXT )
```

Fonte: (MELTON; EISENBERG, 2001)

Após a criação da tabela “information” (Figura 1), podemos implementar os métodos definidos pelo padrão SQL/MM. Um dos métodos mais importantes que o padrão SQL/MM propõe é o CONTAINS, que possibilita realizar diferentes tipos de operações, inclusive as que foram citadas anteriormente. O método CONTAINS verifica se um documento possui ou não determinada palavra ou frase. Esse método pode ser utilizado com outros operadores definidos na ISO/IEC FDC 13249-2 (Figura 2). Por exemplo, ao fazer a junção de dois operadores “SOUNDS” e “LIKE”, obtemos “SOUNDS LIKE”, e podemos realizar consultas por palavras que soam de maneira similar.

Figura 2. Lista de operadores possíveis para a implementação dos métodos definidos pelo padrão SQL/MM.

```
ABOUT | AND | ANY | AS | BROADER | CHARACTERS | ESCAPE  
| EXPAND | FOR | FORM | FROM | FUZZY | IN | IS | LEVEL | LEVELS  
| LIKE | NARROWER | NEAR | NOT | OF | ORDER | PARAGRAPH  
| PARAGRAPHS | PREFERRED | PROXIMITY | RELATED | SAME  
| SENTENCE | SENTENCES | SOUNDS | STEMMED | SYNONYM | TERM  
| THESAURUS | TOP | WITHIN | WORDS
```

Fonte: (ISO/IEC FCD 13249-2 *Information technology - SQL Multimedia and Application Packages - Part 2: Full-Text*, 2002, p. 71)

Considerando a tabela “information” (Figura 1) é possível, por exemplo, executar consultas simples em textos armazenados, que encontram palavras ou frases presentes nos textos, apenas com o uso do método CONTAINS.

O método CONTAINS retorna 1 quando verdadeiro. A primeira consulta (Figura 3), verifica se a tabela “information” contém a palavra “thugs”, já no segundo exemplo (Figura 4), verifica se essa mesma tabela contém a frase “historical documents”.

Figura 3. Consulta ao texto verificando se o mesmo contém a palavra “thugs”, de acordo com o padrão SQL/MM.

```
SELECT docno
FROM information
WHERE information.CONTAINS (“thugs”) = 1
```

Fonte: (MELTON, 2003)

Figura 4. Consulta ao texto verificando se o mesmo contém a frase “historical documents”, de acordo com o padrão SQL/MM.

```
SELECT docno
FROM information
WHERE information.CONTAINS(“historical documents”) = 1
```

Fonte: (MELTON, 2003)

Ainda considerando a tabela “information” (Figura 1) é possível, por exemplo, executar uma consulta identificando palavras derivadas da palavra “standard” e palavras que soam como “sequel”, como mostrado na Figura 5.

Figura 5. Consulta a textos com palavras derivadas de “standard” e com mesma pronúncia de “sequel”, de acordo com o padrão SQL/MM.

```
SELECT docno
FROM information
WHERE information.CONTAINS
      ('STEMMED FORM OF "standard"
      IN SAME PARAGRAPH AS
      SOUNDS LIKE "sequel") = 1
```

Fonte: (MELTON; EISENBERG, 2001)

Na consulta da Figura 5, os operadores STEMMED FORM OF encontrarão as palavras derivadas de “standard” como “standards” e “standardization”, que estão no mesmo parágrafo onde palavras soam como “sequel”, utilizando os operadores SOUNDS LIKE.

Os operadores IS ABOUT determinam se o assunto de um documento está suficientemente relacionado à palavra ou frase especificada na consulta. No exemplo de consulta apresentado na Figura 6, com o uso de IS ABOUT é possível determinar se o documento é sobre “science fiction”.

Figura 6. Consulta a textos usando o IS ABOUT para verificar se o documento é relacionado à “science fiction”, de acordo com o padrão SQL/MM.

```
SELECT docno
FROM information
WHERE information.CONTAINS ('IS ABOUT "science fiction") = 1
```

Fonte: (MELTON, 2003)

Também é possível expandir as consultas por meio de pesquisa difusa, usando o método FUZZY. Esse método inclui as palavras com uma ortografia semelhante ao do termo especificado na pesquisa. Assim, quando, por exemplo, pesquisamos o termo difuso “voila”, resultados com o termo “viola” serão encontrados, apesar do erro ortográfico ocorrido. Além desse, existe o método PROXI, que realiza a pesquisa por

proximidade, encontrando termos que se encontram a uma determinada distância indicada pelo usuário.

As funções de texto completo do SQL/MM permitem consultas que correspondem às palavras ou frases individuais ou a um conjunto de palavras ou frases. Quando é necessário pesquisar documentos usando funções cujos resultados se baseiam nas relações entre palavras ou frases, é possível especificar um dicionário de sinônimos para esse fim. O método SYNONYM, permite a construção de padrões de pesquisa de sinônimos e a pesquisa por ocorrências de sinônimos no texto utilizando um dicionário de sinônimos previamente determinado. Por exemplo, o termo "SQL/MM" é diferente do termo "*SQL Multimedia and Application Packages*", porém, por terem sido definidos como sinônimos, ambos os termos serão incluídos na pesquisa. Outros métodos oferecidos pelo padrão "SQL/MM" também utilizam o dicionário de sinônimos, entre eles destacam-se:

- O método BROADER TERM, que expande a consulta e inclui os termos mais amplos ao termo procurado até que haja uma hierarquia de sinônimos. Por exemplo, a partir da consulta ao termo "SQL" o termo mais amplo "Linguagem de Banco de Dados" será incluído;
- O método TOP TERM, que fornece a construção do padrão de pesquisa de termos principais e a pesquisa de ocorrência dos termos principais associados no texto. Por exemplo, se for definido que "Linguagem de computador" é um termo mais amplo de "Linguagem de banco de dados", o termo principal de "SQL" é "Linguagem de computador";
- Já o método RELATED TERM retorna os termos relacionados ao termo pesquisado e a pesquisa de ocorrências dos termos relacionados associados no texto. Por exemplo, se o termo "cachorro" tem como termo relacionado "lobo", ao pesquisar o termo "cachorro", são retornados todos os documentos que contêm as palavras "cachorro" e "lobo";
- Por fim, destaca-se o método PREFERRED TERM, que permite a pesquisa por termos preferenciais. Por exemplo, podemos definir "SQL" como um termo preferido para "*Structured Query Language*".

Outros dois métodos importantes definidos pelo padrão SQL/MM são: o SCORE, que retorna a medida de relevância das palavras (número de ocorrências da palavra e o número de palavras correspondentes); e o método

NUMBEROFMATCHES que retorna a quantidade de vezes que um padrão de pesquisa é encontrado em um documento.

Além disso, o padrão também inclui a utilização de operadores booleanos nas consultas tais como “&” (e), “|” (ou) e “NOT” (negação), que podem ser utilizados para aumentar ou diminuir o escopo da pesquisa.

É importante salientar que as “*stop words*” também são definidas por padrão, sendo importantes para filtrar as pesquisas e melhorar o desempenho. As “*stop words*” são basicamente conjuntos de palavras que ocorrem com tanta frequência, que se tornam inúteis para fins de identificação do texto, não influenciam a interpretação do documento, não sendo consideradas na consulta (por exemplo os artigos – o, a, os, as).

3. ESTUDO DOS SGBD

Como abordado no Capítulo 2, os SGBD objeto-relacionais que atendem ao padrão SQL/MM possuem recursos para o tratamento de dados multimídia, sendo que alguns desses SGBD podem tratar do tipo de dado *Full Text*. Desse modo, este capítulo apresenta uma análise dos principais SGBD quanto ao tratamento do tipo de dado *Full Text*.

3. 1. Análise dos principais SGBD Objeto-Relacionais em relação aos recursos propostos pelo padrão SQL/MM

Para o estudo, inicialmente foi feita uma pré-seleção de cinco SGBD muito utilizados atualmente: MySQL, PostgreSQL, Oracle, DB2 e SQL Server. Ao analisar as documentações de cada um deles, verificou-se quais atendem às especificações que o padrão SQL/MM determina para o dado do tipo *Full Text*, tais como: a busca por palavras específicas, a busca por palavras derivadas, com significados similares, e a busca por palavras que são pronunciadas de maneira semelhante. Tendo em vista essas especificações, conforme apresentado no Quadro 1, os SGBD MySQL e PostgreSQL, não oferecem essas funcionalidades esperadas aos tipos de dados *Full Text* definidas pelo padrão SQL/MM. Portanto, a análise seguiu considerando apenas os SGBD Oracle, DB2 e o SQL Server, demonstrando quais são as especificações atendidas.

Quadro 1. Suporte ao tipo de dados *Full Text* pelos SGBD pré-selecionados.

	MySQL	PostgreSQL	Oracle	DB2	SQL Server
<i>Full Text</i>	Não	Não	Sim	Sim	Sim

Fonte: Desenvolvido pelo autor

Antes disso, é importante destacar que os SGBD estudados utilizam nomes diferentes, ou até mesmo implementações diferentes, para funções muito semelhantes. Por isso, com o intuito de facilitar a visualização, temos o Quadro 2 que apresenta as funcionalidades definidas pelo padrão SQL/MM que são oferecidas pelos SGBD Oracle, SQL Server e DB2.

Quadro 2. Comparação entre os SGBD Oracle, SQL Server e DB2 em relação ao suporte às funcionalidades definidas pelo padrão SQL/MM.

Funcionalidade	Oracle	SQL Server	DB2
Pesquisa por termo simples ou frase	Sim, realizada a partir do método CONTAINS.	Sim, podendo ser realizada a partir do predicado CONTAINS e da função CONTAINSTABLE.	Sim, realizada a partir do método CONTAINS e da função XMLCOLUMN-CONTAINS.
Pesquisa por prefixo	Sim, utiliza o caractere asterisco (*) para essa finalidade, mas não é específica para Oracle Text.	Sim, utilizando o asterisco (*) antes de uma palavra ou frase.	Sim, utilizando o asterisco (*) antes de uma palavra ou frase.
Pesquisa por palavras derivadas	Sim, usando o operador STEM.	Sim, usando a cláusula FORMS OF junto com a palavra INFLECTIONAL.	Não.
Pesquisa por proximidade	Sim, com a utilização do operador NEAR.	Sim, com a utilização do termo de proximidade NEAR ou o til (~).	Sim, usando o símbolo til (~) e especificando a distância entre as palavras com um número inteiro válido.
Pesquisa por sinônimos	Sim, criando um dicionário de sinônimos e usando o operador SYN.	Sim, criando um dicionário de sinônimos e o usando por meio da cláusula FORMS OF com a palavra THESAURUS.	Sim, criando um dicionário de sinônimos e usando o operador THESAURUS.
Pesquisa difusa	Sim, utilizando o operador FUZZY.	Não.	Sim, usando o símbolo til (~) no final de um termo.

Continua

Funcionalidade	Oracle	SQL Server	DB2
Pesquisa por palavras que soam parecido	Sim, utilizando o operador SOUNDEX.	Sim, mas não é específica para <i>Full-Text Search</i> .	Sim, mas não é específica para <i>Search Text</i> .
Pesquisa por termo mais amplo	Sim, usando o operador BROADER TERM.	Não.	Não.
Pesquisa pelos principais termos	Sim, usando o operador TOP TERM.	Não.	Não.
Pesquisa por termos relacionados	Sim, usando o operador RELATED TERM.	Não.	Não.
Pesquisa por termos preferenciais	Sim, usando o operador PREFERRED TERM.	Não.	Não.
Medida de relevância	Sim, com o operador SCORE	Sim, com o uso do SCORE.	Sim, com a função SCORE.
Stop words	Sim.	Sim.	Sim.

Fonte: Desenvolvido pelo Autor

O Quadro 2 demonstra as funcionalidades oferecidas pelos três SGBD (Oracle, SQL Server e DB2) que estão de acordo com o padrão SQL/MM. Nele podemos observar que os três gerenciadores possuem funcionalidades em comum, tais como: pesquisa simples por palavras ou frases, pesquisa por prefixo, pesquisa por palavras que soam semelhante, obtém a medida de relevância, e as “*stop words*”, lista de palavras irrelevantes.

A pesquisa difusa é realizada no Oracle utilizando o operador FUZZY, já no DB2 é feita utilizando o caractere til (~) no final da palavra pesquisada. Apenas o Oracle e o SQL Server possuem a pesquisa por palavras derivadas.

Ainda que todos os gerenciadores sejam capazes de suportar o dicionário de sinônimos, operações como pesquisa por termo mais amplos, pesquisa pelos principais termos, pesquisa por termos relacionados e pesquisa por termos preferenciais, são oferecidas apenas pelo Oracle.

3. 2. Estudo mais aprofundado dos SGBD Objeto-Relacionais Oracle, SQL Server e DB2

Tendo em vista que foram apresentados os recursos oferecidos pelos SGBD Oracle, SQL Server e DB2 relacionados ao padrão SQL/MM, é possível analisar mais profundamente as suas funcionalidades e exemplos de implementações.

3. 2. 1. DB2

O DB2 possui o mecanismo *Text Search*, que permite ao SGBD pesquisas a tipos de dados *Full Text* de vários formatos de documento, como: texto, HTML e XML. O funcionamento do *Text Search*, se dá a partir da inicialização de uma instância DB2 *Text Search* e da criação de índices *Text Search*, que possibilita a realização de consultas aos textos por meio de instruções SQL e XQUERY (linguagem similar ao SQL, projetada para pesquisas em documentos XML).

A Figura 7 representa a criação da tabela “noticia” com as colunas “id_noticia”, “titulo” e “descrição”, sendo que a coluna “descrição” foi definida como XML (linguagem nativa do DB2).

Figura 7. Criação de uma tabela “noticia” no DB2.

```
CREATE TABLE noticia (  
    id_noticia INTEGER NOT NULL,  
    titulo VARCHAR(20) NOT NULL,  
    descrição XML,  
    PRIMARY KEY (id_noticia)  
);
```

Fonte: Desenvolvido pelo autor

Após a criação da tabela, é possível realizar a indexação, sendo de fundamental importância para uma rápida recuperação dos documentos mais relevantes à pesquisa. Na Figura 8, a indexação foi feita na coluna “descrição” da tabela “noticia” com o seguinte comando:

Figura 8. Criação de índice na coluna “descrição” no DB2.

```
CREATE INDEX indice_noticia FOR TEXT ON noticia(descrição)
```

Fonte: Adaptado de IBM (S/D).

O DB2 oferece funções de pesquisa para texto completo integradas, como CONTAINS e a XMLCOLUMN-CONTAINS, que podem ser usadas para consultas a palavras ou frases específicas, e o SCORE.

A função XMLCOLUMN-CONTAINS, retorna os documentos XML que estejam de acordo com os termos de pesquisa especificados. Enquanto a função CONTAINS, procura pelo índice de texto que esteja de acordo com o especificado e retorna um valor que indica se uma correspondência foi ou não encontrada. A partir dessas funções, é possível executar as pesquisas previstas pelo SQL/MM abordadas no Capítulo 2.

A Figura 9 apresenta os comandos para a realização da pesquisa simples no DB2. A função CONTAINS é utilizada para que seja retornado 1, caso a correspondência da palavra “coronavírus” seja encontrada na “descrição”, e 0, caso ocorra ao contrário. Já na Figura 10 é apresentado um exemplo bem semelhante, porém, com a adição da função SCORE de modo a obter também a medida de relevância.

Figura 9. Pesquisa simples usando o método CONTAINS para encontrar a palavra “coronavírus” na coluna “descrição” no DB2.

```
SELECT titulo, descrição
FROM noticia
WHERE CONTAINS(descrição, 'coronavírus') = 1
```

Fonte: Adaptado de IBM (S/D).

Figura 10. Pesquisa simples por palavra usando o método CONTAINS para encontrar a palavra “coronavírus” na coluna “descrição” e o SCORE para determinar a medida relevância dessa palavra no DB2.

```
SELECT titulo, descrição
FROM noticia
WHERE CONTAINS(descrição, 'coronavírus') = 1
ORDER BY SCORE(descrição, 'coronavírus') DESC
```

Fonte: Adaptado de IBM (S/D).

3. 2. 2. ORACLE

O Oracle possui o *Oracle interMedia* - um conjunto de ferramentas que permitem o armazenamento e a manipulação de dados multimídia como textos, imagens, áudios e vídeos, por exemplo - que contém o mecanismo *Oracle Text* (integrado no *Oracle Database 12c* (todas as edições)), responsável pela recuperação de texto completo, em arquivos ou na web. O *Oracle Text* também suporta diferentes formatos de documento em suas aplicações, como textos não estruturados, PDF, HTML e XML, e oferece suporte de pesquisa à vários idiomas ocidentais (inglês, francês, espanhol, alemão etc.) e orientais (japonês, coreano, chinês tradicional e simplificado), oferecendo o melhor conjunto multilinguístico do mercado (FORD, 2017).

A Figura 11 apresenta a utilização do operador TRSYN, que localiza as notícias cuja “descrição” contém a palavra “vírus” ou seus equivalentes no idioma chinês.

Figura 11. Localiza as notícias com a descrição “vírus” ou seus equivalentes no idioma chinês no Oracle.

```
SELECT id_noticia, titulo
FROM noticia
WHERE CONTAINS(descrição, 'TRSYN(vírus, Chinese)')
```

Fonte: Adaptado de Roger Ford (2017)

No exemplo da Figura 11, o operador de termo de tradução ‘TRSYN’, necessita do carregamento de um dicionário de sinônimos de tradução. Apesar do Oracle não possuir o seu próprio dicionário de sinônimos, ele dá suporte a esse tipo de carregamento.

A Figura 12 demonstra a criação de uma tabela “noticia” no Oracle, com as colunas “id_noticia”, “titulo” e “descrição” para que posteriormente seja possível a realização de consultas aos dados da tabela.

Figura 12. Criação de uma tabela no Oracle.

```
CREATE TABLE noticia (
id_noticia PRIMARY KEY,
titulo VARCHAR2 (100),
descrição VARCHAR2 (200)
);
```

Fonte: Desenvolvido pelo autor

O *Oracle Text* oferece três tipos de índices: CONTEXT, CTXCAT e CTXRULE, que, por sua vez, atendem às necessidades de pesquisa de texto. O índice CONTEXT, com o uso do operador CONTAINS, é adequado para a indexação de coleções de documentos grandes e coerentes. O índice CTXCAT, por meio do operador CATSEARCH, é utilizado para indexação de fragmentos curtos de texto. Por último, o índice CTXRULE, com o operador MATCHES, realiza consultas baseadas em regras de classificação definidas pelo usuário. Entretanto, essa última abordagem funciona apenas para conjuntos pequenos de dados. Para grandes conjuntos de

dados, o *Oracle Text* oferece um treinamento de classificação, no qual o pacote CTX_CLS.CLUSTERING, permite a criação de clusters, que a partir de amostras de conjuntos de documentos classificadas pelo usuário, agrupam automaticamente um conjunto de documentos de acordo com seus significados semânticos.

A Figura 13 apresenta a criação do índice CONTEXT com o nome “indice_noticia” na coluna “descrição” da tabela “noticia”.

Figura 13. Criação de um índice CONTEXT para a coluna “descrição” no Oracle.

```
CREATE INDEX indice_noticia ON  
noticia (descrição)  
INDEXTYPE IS CTXSYS.CONTEXT
```

Fonte: Adaptado de Oracle (S/D)

Além dos recursos tradicionais de pesquisa de texto previstos no padrão SQL/MM (Capítulo 2), o *Oracle Text* oferece o operador ABOUT, que obtém os documentos relevantes dentro de um tema específico (por exemplo, o uso de um ABOUT em uma consulta com a palavra “Paraná” poderá retornar documentos que contenham os termos “Londrina” e “Curitiba”, que são nome de cidades situadas no Paraná). Também é possível atribuir pesos aos termos da consulta por meio do caractere (*). Com o (*) o usuário pode ponderar o(s) termo(s) da consulta por um valor que varia de acordo com a importância que aquele(s) termo(s) tem na consulta, resultando em documentos que melhor refletem seu interesse.

As figuras 14 e 15 demonstram a utilização do operador ABOUT. No primeiro exemplo (Figura 14), consegue-se localizar todos os documentos sobre “futebol”.

Figura 14. Uso do operador ABOUT para encontrar documentos sobre “futebol” no Oracle.

```
SELECT titulo FROM noticia  
WHERE CONTAINS (descrição, 'ABOUT(futebol)')
```

Fonte: Adaptado de Oracle (S/D)

No segundo exemplo (Figura 15), é possível refinar a pesquisa e encontrar os documentos sobre as regras de futebol em competições internacionais. Os documentos retornados serão “futebol”, “regras” ou “competição internacional”, sendo que os documentos que obtiverem a classificação de relevância mais alta, por meio do SCORE, serão aqueles que apresentarem esses três temas juntos. Esse valor diminui de acordo com a quantidade de termos encontrados.

Figura 15. Uso da operação ABOUT para encontrar documentos sobre “regras de futebol em competições internacionais” no Oracle.

```
SELECT SCORE(1), titulo FROM noticia
WHERE CONTAINS (descrição, 'ABOUT(regras de futebol em competições
internacionais)', 1) > 0
ORDER BY SCORE(1) DESC;
```

Fonte: Adaptado de Oracle (S/D)

Além do que foi mencionado, o *Oracle Text*, oferece mais três recursos de pesquisa de texto: *Highlighting*, *Markup* e *Snippet*, e três recursos linguísticos avançados: *Gist Generation* e *Theme Extraction*, que promovem uma melhor experiência de visualização aos usuários.

Os quadros 3 e 4, resumem como funcionam cada um dos recursos mencionados anteriormente.

Quadro 3. Recursos *Highlighting*, *Markup* e *Snippet* no Oracle

Recurso	Ação	Operador
<i>Highlighting</i>	Destaca as palavras do documento que satisfazem à pesquisa. Com ele é possível, por exemplo, destacar as palavras do documento atendam a consulta ou destacar as palavras e frases que melhor representam a consulta usando o operador ABOUT.	CTX_QUERY.HIGHLIGHT
<i>Markup</i>	Além de obter as palavras que satisfazem à pesquisa como o <i>Highlighting</i> , o <i>Markup</i> produz uma versão do texto com essas palavras marcadas (entre <> ou entre elementos do HTML como)	CTX_DOC.MARKUP
<i>Snippet</i>	Produz um texto com as palavras-chave destacadas dando ao usuário noção do que se trata o texto antes mesmo de abri-lo.	CTX_DOC.SNIPPET

Fonte: Desenvolvido pelo autor

Quadro 4. Recursos linguísticos *Gist Generation* e *Theme Extraction* no Oracle

Recurso	Ação	Operador
<i>Theme Extraction</i>	Permite ao usuário a pesquisa por assunto, mesmo que o documento não tenha mencionado o mesmo explicitamente. Retorna uma lista de documentos e a sua pontuação de relevância.	CTX_DOC.THEMES
<i>Gist Generation</i>	Fornece um sumário que contém frases ou parágrafos que melhor representam o conteúdo do documento que está sendo analisado.	CTX_DOC.GIST

Fonte: Desenvolvido pelo autor

Enquanto o *Gist Generation* fornece um sumário que descreve sobre o que é o documento, o *Theme Extraction* exibe uma lista de documentos que melhor representam o assunto pesquisado.

3. 2. 3. SQL SERVER

O SQL Server possui o mecanismo de pesquisa *Full-Text Search* para realização da pesquisa a dados do tipo *Full Text*. Os tipos de formatos de documentos suportados são: PDF, HTML e XML. Antes de iniciar as pesquisas propriamente ditas, são necessárias duas etapas: a criação de um catálogo de índice *Full Text* (sua criação é opcional, quando não é criado, o catálogo padrão será usado) e a criação dos índices *Full Text*. O catálogo nada mais é do que um agrupamento de índices *Full Text*, desenvolvido com objetivo de facilitar na manutenção. Já os índices *Full Text* são essenciais para realizar as consultas a texto completo.

A Figura 16 apresenta a criação da tabela “noticia” no SQL Server com as seguintes colunas: “id_noticia”, “titulo” e “descrição”. E as figuras 17 e 18, respectivamente, demonstram os comandos para a criação de um catálogo “ftCatalogo” e de um índice na tabela “noticia”.

Figura 16. Criação de uma tabela no SQL Server.

```
CREATE TABLE noticia (  
    id_noticia INT,  
    titulo VARCHAR(100),  
    descrição VARCHAR(200),  
    PRIMARY KEY(id_noticia)  
)
```

Fonte: Adaptado de Microsoft (2017)

Figura 17. Comando para criar um catálogo no SQL Server.

```
CREATE FULLTEXT CATALOG ftCatalogo
```

Fonte: Adaptado de Microsoft (2017)

Figura 18. Comando para criar um índice no SQL Server.

```
CREATE FULLTEXT INDEX ON noticia (descrição)
```

Fonte: Adaptado de Microsoft (2017)

Depois da criação dos índices, os usuários podem executar consultas no texto completo armazenado em colunas de tabelas. Para facilitar esse processo, o SQL Server oferece os predicados CONTAINS e FREETEXT, e as funções CONTAINSTABLE e FREETEXTTABLE, usados com a instrução SELECT.

CONTAINS e CONTAINSTABLE analisam as palavras e frases morfologicamente e podem ser usados juntamente com operadores booleanos (*and*, *or* e *not*), permitindo combinar termos da pesquisa. Eles retornam as linhas (se houver) que atendam aos critérios de pesquisa especificados.

FREETEXT e FREETEXTTABLE analisam com significado e dentro de um contexto específico. Retornam as linhas (se houver) que atendam aos critérios de seleção especificados na pesquisa (ocorrência). É importante salientar que as funções CONTAINSTABLE e FREETEXTTABLE retornam o valor de classificação de relevância (RANK), e a chave de texto completo (KEY), exclusiva para cada linha.

Por meio do uso desses predicados e funções, é possível realizar diferentes tipos de consultas em dados *Full Text*, alguns deles abordados no Capítulo 2. Além desses, o SQL Server oferece a pesquisa por termo ponderado (também presente no Oracle, mas com implementação diferente), sendo realizada juntamente com a função CONTAINSTABLE e o operador ISABOUT, na qual o usuário especifica um valor de 0.0 (valor mais baixo) a 1.0 (valor mais alto), indicando a importância de uma palavra ou frase nos conjuntos de palavras e frases. O valor da classificação (RANK), utilizando o ISABOUT, é feito por meio do algoritmo de Jaccard (MICROSOFT, 2017).

As figuras 19 e 20 mostram exemplos de consultas a dados do tipo *Full Text*. Na Figura 19 é apresentada uma consulta por termo simples no SQL Server para localizar todas as descrições de notícia que possuem a frase “learning curve” usando apenas o operador CONTAINS. Já na Figura 20, a consulta é realizada para encontrar na coluna “descrição” palavras que possuem o prefixo “top”.

Figura 19. Consulta por termo simples no SQL Server para localizar todas as descrições de notícias que possuem a frase “learning curve”.

```
SELECT titulo, descrição  
FROM noticia  
WHERE CONTAINS(descrição, '" learning curve"')
```

Fonte: Adaptado de Microsoft (2017)

Figura 20. Consulta por termo de prefixo no SQL Server para localizar todos as descrições que contenham o prefixo “top”.

```
SELECT id_noticia, descrição  
FROM noticia  
WHERE CONTAINS (descrição, '" top* "')
```

Fonte: Adaptado de Microsoft (2017)

No exemplo da Figura 20, palavras que comecem com o termo “top”, especificado antes do asterisco (*), serão retornadas. Se a delimitação não for feita por aspas duplas, a pesquisa de texto completo não considerará o asterisco como sendo curinga.

O SQL Server oferece também funções de pesquisa semântica, que promovem um conhecimento mais profundo sobre os documentos não estruturados armazenados em bancos de dados. A pesquisa semântica foi um recurso introduzido no SQL Server desde o SQL Server 2012, e a partir dela é possível encontrar palavras estatisticamente relevantes (palavras-chave), bem como localizar documentos semelhantes entre si com base nessas palavras-chave. Porém, antes de executar as consultas, é necessária a habilitação dos recursos necessários para a realização da pesquisa semântica. A Figura 21 mostra a criação de um índice semântico na coluna “descrição”.

Figura 21. Criação de um índice semântico na coluna “descrição”

```
CREATE FULLTEXT INDEX ON noticia
    (descrição
      Language 1046 Statical_Semantics
    )
KEY INDEX id_noticia
WITH STOPLIST = SYTEM
```

Fonte: Adaptado de Microsoft (2017)

As funções de pesquisa semântica são três: SEMANTICKEYPHRASETABLE, SEMANTICSIMILARITYDETAILSTABLE e SEMANTICSIMILARITYTABLE. A função SEMANTICKEYPHRASETABLE retorna as linhas que possuem as palavras-chave especificadas, ou seja, o usuário pode obter todas as linhas que possuem a palavra “coronavírus” em seu conteúdo, por exemplo. Outra funcionalidade que essa função permite é a de localizar as palavras-chave em um documento com o respectivo SCORE. Já a função SEMANTICSIMILARITYTABLE obtém os documentos que foram identificados como semelhantes ou relacionados ao documento, assim, o usuário pode determinar uma linha como referência e visualizar o grau de similaridade de todas as linhas em relação a esta. Por último, a função SEMANTICSSIMILARITYDETAILSTABLE pode ser usada para localizar as palavras-chave comuns em dois documentos.

3. 3. Resultados e conclusões

Com o objetivo de comparar e demonstrar as diferenças e similitudes dos SGBD DB2, Oracle e SQL Server em relação ao tratamento de tipos de dados *Full Text*, suas funcionalidades e limitações foram relacionadas no Quadro 5. As características consideradas foram: tipos de dados suportados, formatos de documentos, idiomas e alguns tipos de pesquisa adicionais abordados nos tópicos anteriores.

Quadro 5. Comparação dos SGBD DB2, Oracle e SQL Server em relação aos tipos de dados, formatos de documentos, idiomas e tipos de pesquisa.

	DB2	Oracle	SQL Server
Nome do componente	<i>Search Text</i>	<i>Oracle Text</i>	<i>Full Text Search</i>
Tipos de dados	CHAR, VARCHAR, NCHAR, NVARCHAR, TEXT, NTEXT, XML e VARBINARY	VARCHAR (até 4000 caracteres), CLOB, BLOB	CHAR, VARCHAR, NCHAR, NVARCHAR, TEXT, NTEXT, IMAGE, XML OU VARBINARY(MAX) e FILESTREAM
Formatos de documentos	Texto, HTML e XML	PDF, HTML e XML	PDF, HTML e XML
Idiomas	Árabe, tcheco, dinamarquês, alemão, inglês, espanhol, francês, italiano, japonês, coreano, holandês, polonês, português, russo, sueco, chinês	Idiomas ocidentais (inglês, francês, espanhol, alemão, etc...) e orientais (japonês, coreano, chinês tradicional e simplificado)	Suporte a todos os idiomas que têm suporte nos sistemas operacionais Windows (quase 50 idiomas diferentes)
Tipos de pesquisas adicionais	Não tem	Pesquisa dentro de um tema específico Pesquisa por termo ponderado Oferece recursos linguísticos avançados que retornam, por exemplo, um resumo do que se trata o texto	Pesquisa por termo ponderado Pesquisa semântica

Fonte: Desenvolvido pelo autor

A partir das comparações realizadas, concluiu-se que o DB2, apesar de possuir algumas das funcionalidades presentes no padrão SQL/MM, não apresenta recursos adicionais de pesquisa, ficando aquém dos outros SGBD. Por outro lado, o Oracle e

o SQL Server, além de atenderem uma gama maior de funcionalidades previstas pelo padrão, possibilitam pesquisas adicionais e mais amplas, que vão desde análises morfológicas das palavras até pesquisas com significado.

Ambos os SGBD possuem recursos avançados de pesquisa a tipos de dados *Full Text*. Porém, quando se trata de usabilidade e disponibilidade, o SQL Server se destaca, tendo em vista que oferece mais materiais explicativos sobre o uso da ferramenta disponíveis na web. Ademais, o SQL Server é disponibilizado gratuitamente aos alunos da UNICAMP.

Desse modo, o SQL Server foi escolhido para dar continuidade ao desenvolvimento deste trabalho, sendo utilizado na realização de um estudo de caso que visa a aplicação de pesquisa semântica, bem como outros recursos relacionados a tipos de dados *Full Text*, na avaliação de questões discursivas.

4. ESTUDO DE CASO: COMO A PESQUISA SEMÂNTICA PODE AJUDAR NA AVALIAÇÃO DE QUESTÕES DISCURSIVAS?

Após o estudo das funcionalidades oferecidas pelo SGBD escolhido, o SQL Server, para a realização de pesquisas semânticas e tratamento do tipo de dado *Full Text*, a sua aplicabilidade na avaliação de questões discursivas foi avaliada por meio de um estudo de caso. A versão utilizada foi o SQL Server 2017.

Para o desenvolvimento do estudo de caso, criou-se um banco de dados, onde foram inseridas as seguintes respostas para a questão **“Explique a direção dos ventos alísios no Atlântico Sul e a sua funcionalidade no transporte marítimo da África para o Brasil”** retirada da prova da segunda fase da UNICAMP 2020 da área Ciências Humanas:

(RESPOSTA DO PROFESSOR)

A direção dos ventos alísios no litoral brasileiro é de sudeste para noroeste; no litoral africano, é de noroeste para sudeste. O desenvolvimento das técnicas de navegação naquele contexto estava relacionado ao aproveitamento do regime dos ventos, dados que as embarcações eram movidas a vela. Nesse sentido, a navegação da África para o Brasil era facilitada.

(RESPOSTA INCORRETA)

O comércio de pessoas escravizadas desempenhou papel fundamental na dinamização de vários tipos de trocas estabelecidas entre o Brasil e África no contexto da colonização. Por meio de uma multiplicidade de relações potencializadas pelo tráfico, formou-se um amplo complexo histórico e cultural afro-brasileiro.

(RESPOSTA INCOMPLETA)

Os ventos alísios, que sopram no Hemisfério Sul na proximidade do Trópico de Capricórnio em direção ao Equador, sofrem inclinação devido à rotação da Terra (Efeito Coriolis) e adquirem direção aproximada oeste-leste atuando durante o ano todo. Como antigamente as embarcações eram movidas à vela, os ventos facilitavam e permitiam que o transporte marítimo ocorresse.

(RESPOSTA CORRETA)

A direção dos ventos alísios no litoral brasileiro são do sudeste para o noroeste, enquanto isso, no litoral africano são do noroeste para o sudeste. Como antigamente as embarcações eram movidas à vela, a direção dos ventos era de extrema importância para a navegação. Dessa forma, a direção dos ventos contribua para que o transporte marítimo da África para o Brasil fosse facilitado.

(RESPOSTA MEIO CERTA)

A direção dos ventos alísios no litoral brasileiro são do noroeste para o sudeste, enquanto isso, no litoral africano são do sudeste para o noroeste. Como antigamente as embarcações eram movidas à vela, a direção dos ventos era de extrema importância para a navegação. Dessa forma, a direção dos ventos contribua para que o transporte marítimo da África para o Brasil fosse facilitado.

As respostas foram desenvolvidas de modo que deixassem o mais claro possível as suas peculiaridades: certo, meio certa, incompleta e incorreta. A “Resposta do Professor”, representa o que o avaliador gostaria de obter como resposta, tendo sido retirada do banco de provas da Unicamp. A “Resposta Incorreta”, apresenta palavras em comum com a “Resposta do Professor”, mas não responde à pergunta. A “Resposta incompleta”, responde apenas parcialmente a pergunta e possui partes fora do contexto esperado, sendo irrelevantes para a resposta. A “Resposta Correta”, responde à pergunta abordando todos os pontos esperados na “Resposta do Professor”. E na “Resposta meio certa”, o aluno inverteu a direção dos ventos, dizendo que no Brasil, os ventos vão na direção noroeste para sudeste, e na África vão do sudeste para o noroeste. A Figura 22 exibe a tabela, denominada “BD_Resposta”, que armazena o número identificador da resposta (coluna “DocId”), a classificação da resposta (coluna “Questão”) e o texto da resposta propriamente dito (coluna “Resposta”).

Figura 22. Exibição da tabela do banco de dados criado.

DocId	Questão	Resposta
1	Resposta do Professor	A direção dos ventos alísios no litoral brasileiro é de sudeste para noroeste; no litoral africano, é de n...
2	Resposta incorreta	O comércio de pessoas escravizadas desempenhou papel fundamental na dinamização de vários tip...
3	Resposta incompleta	Os ventos alísios, que sopram no Hemisfério Sul na proximidade do Trópico de Capricórnio em direçã...
4	Resposta correta	A direção dos ventos alísios no litoral brasileiro são do sudeste para o noroeste, enquanto isso, no lit...
5	Resposta meio certa	A direção dos ventos alísios no litoral brasileiro são do noroeste para o sudeste, enquanto isso, no lit...

Fonte: Desenvolvido pelo autor

O *script* que mostra todo o processo de criação de tabelas, indexação e habilitação dos recursos necessários para a realização de consultas com o uso da pesquisa semântica, estão disponíveis no APÊNDICE A deste trabalho.

Conforme abordado na seção 3. 2. 3., após a criação do catálogo *Full Text*, do índice *Full Text* e de ter habilitado a pesquisa semântica é possível realizar as consultas. Analisaremos não apenas a funcionalidade, mas daremos exemplos de onde elas podem ser aplicadas na avaliação de questões discursivas, levando em consideração o banco de dados criado.

Para uma melhor organização das informações, pontuamos duas possibilidades de análise:

- (1) o avaliador informa as palavras mais relevantes para a sua análise;
- (2) respostas que o SQL Server compara automaticamente sem a necessidade de especificar palavras.

4. 1. PRIMEIRA POSSIBILIDADE DE ANÁLISE: O AVALIADOR INFORMA AS PALAVRAS RELEVANTES

A) O AVALIADOR DETERMINA AS PALAVRAS RELEVANTES E ATRIBUI PESOS PARA OBTER A CLASSIFICAÇÃO DAS RESPOSTAS

Com o uso da função *CONTAINSTABLE* aplicada a tipos de dados *Full Text* é possível obter as linhas ranqueadas a partir da coluna *RANK*.

A coluna *RANK* pode ter valores variando de 0 a 1000, que indica o grau de correspondência das linhas em relação aos critérios de seleção, de forma que quanto maior o valor da classificação, maior o grau de correspondência entre os resultados.

Quando utilizamos o *CONTAINSTABLE* com outros operadores, pesquisas mais personalizadas são possíveis. Por exemplo, ao usar os operadores *ISABOUT* e

WEIGHT, o avaliador consegue ponderar as palavras/frases de acordo com a sua relevância. Na Figura 23, a consulta retorna todas as respostas que contêm as palavras/frases “Sudeste”, “Noroeste”, “Ventos alísios” e “Navegação” com os seus respectivos pesos. Os pesos, ficam à critério do avaliador e podem variar entre 0.0 a 1.0.

Figura 23. Consulta a termos ponderados no SQL Server.

```

SELECT FT_TBL.Questão, KEY_TBL.RANK, FT_TBL.Resposta
FROM BD_Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(BD_Resposta, Resposta,
        'ISABOUT ("Sudeste" WEIGHT (.1),
        "Noroeste" WEIGHT (.1), "Ventos alísios" WEIGHT (.1), "Navegação"
        WEIGHT (.2))' ) AS KEY_TBL
    ON FT_TBL.DocId = KEY_TBL.[KEY]
WHERE FT_TBL.DocId > 1 -- retirando a resposta do professor
ORDER BY KEY_TBL.RANK DESC

```

Fonte: Desenvolvido pelo autor

A Figura 24 apresenta os resultados obtidos na consulta da Figura 23. Observa-se que a “Resposta incompleta” é a que possui menor classificação (RANK 23), enquanto que a “Resposta correta” e a “Resposta meio certa” apresentaram o mesmo RANK com o valor de 122. A “Resposta incorreta” não pontuou.

Figura 24. Resultados obtidos na consulta da Figura 23.

Questão	RANK	Resposta
Resposta correta	122	A direção dos ventos alísios no litoral brasileiro sã...
Resposta meio certa	122	A direção dos ventos alísios no litoral brasileiro sã...
Resposta incompleta	23	Os ventos alísios, que sopram no Hemisfério Sul ...

Fonte: Desenvolvido pelo autor

A partir do uso do WEIGHT (Figura 23), o avaliador consegue não apenas a classificação das respostas de acordo com as palavras-chave utilizadas, mas também pode incrementar a consulta determinando pesos diferentes para essas

palavras/frases de acordo com seus critérios de avaliação. É importante salientar, que caso o aluno repita uma palavra/frase ponderada, a classificação de sua resposta aumenta, tendo, portanto, a necessidade de pré-processamento para filtrar palavras/frases repetidas.

B) O AVALIADOR DETERMINA AS PALAVRAS-CHAVE NECESSÁRIAS NAS RESPOSTAS E OBTÉM O SCORE

Caso o avaliador queira determinar as palavras-chave que são necessárias nas respostas e obter a estatística de significância de cada palavra no documento (SCORE), ele pode utilizar a função de pesquisa semântica SEMANTICKEYPHRASETABLE. No exemplo da Figura 25, as linhas com as respostas que possuem as palavras “Sudeste”, “Noroeste” ou “Ventos”, serão retornadas juntamente com o SCORE que elas apresentam. É importante salientar que quando utilizamos funções de pesquisa semântica, apenas palavras podem ser consultadas, isso porque apenas palavras isoladas são indexadas, não sendo possível a indexação de frases (n-gramas) (MICROSOFT, 2017). Isso explica o motivo que na consulta da Figura 25 não foi possível especificar a frase “Ventos alísios”, diferente do que aconteceu no exemplo anterior (Figura 23), o qual não utilizou funções de pesquisa semântica.

Figura 25. Consulta as respostas que possuem as palavras “Sudeste”, “Noroeste” ou “Ventos” no SQL Server.

```
SELECT  s.keyphrase, s.SCORE, d.DocId, d.Questão, d.Resposta
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
      INNER JOIN BD_Resposta AS d
      ON s.document_key = d.DocId
WHERE s.keyphrase in ('Sudeste', 'Noroeste', 'Ventos') and d.DocId > 1
ORDER BY s.SCORE DESC
```

Fonte: Desenvolvido pelo autor

A Figura 26 apresenta os resultados obtidos na consulta da Figura 25. Como podemos observar, as palavras “Noroeste” e “Sudeste”, apresentaram relevância máxima nas respostas correta e meio certa.

Figura 26. Resultados obtidos na consulta da Figura 25.

	keyphrase	score	Questão	Resposta
1	noroeste	1	Resposta correta	A direção dos ventos alísios no litoral brasileiro sã...
2	sudeste	1	Resposta correta	A direção dos ventos alísios no litoral brasileiro sã...
3	noroeste	1	Resposta meio certa	A direção dos ventos alísios no litoral brasileiro sã...
4	sudeste	1	Resposta meio certa	A direção dos ventos alísios no litoral brasileiro sã...
5	ventos	0,3652068	Resposta meio certa	A direção dos ventos alísios no litoral brasileiro sã...
6	ventos	0,3652068	Resposta correta	A direção dos ventos alísios no litoral brasileiro sã...
7	ventos	0,3652068	Resposta incompleta	Os ventos alísios, que sopram no Hemisfério Sul ...

Fonte: Desenvolvido pelo autor

C) O AVALIADOR DETERMINA AS PALAVRAS-CHAVE NECESSÁRIAS EM UMA RESPOSTA E OBTÉM O SCORE

Caso o avaliador queira verificar se as palavras desejadas estão ou não presentes em uma resposta específica, basta apenas passar um terceiro parâmetro na função de pesquisa semântica SEMANTICKEYPHRASETABLE, indicando o identificador (chave primária da linha) da resposta desejada. Dessa maneira, assim como apresentado na Figura 27, as palavras-chave presentes na “Resposta correta” (DocId = 4) serão retornadas com os seus respectivos SCOREs.

Figura 27. Consulta ao SCORE das palavras “Sudeste”, “Noroeste” e “Ventos” na “Resposta correta” (DocId = 4).

```

SELECT  s.keyphrase, s.SCORE, d.Questão, d.Resposta
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta, 4) AS s
      INNER JOIN BD_Resposta AS d
      ON s.document_key = d.DocId
WHERE s.keyphrase in ('Sudeste', 'Noroeste', 'Ventos')
ORDER BY s.SCORE DESC

```

Fonte: Desenvolvido pelo autor

A Figura 28 apresenta os resultados obtidos na consulta da Figura 27, onde podemos visualizar o SCORE das palavras “Sudeste”, “Noroeste” e “Ventos” na “Resposta correta”.

Figura 28. Resultados obtidos na consulta da Figura 27.

	keyphrase	score	Questão	Resposta
1	noroeste	1	Resposta correta	A direção dos ventos alísios no litoral brasilei...
2	sudeste	1	Resposta correta	A direção dos ventos alísios no litoral brasilei...
3	ventos	0,3652068	Resposta correta	A direção dos ventos alísios no litoral brasilei...

Fonte: Desenvolvido pelo autor

Mesmo o SQL Server possibilitando tais funcionalidades, ele possui a limitação de não permitir a pesquisa por frases nas funções de pesquisa semântica. Na tentativa de contornar essa limitação, podemos utilizar o predicado CONTAINS, que permite que o avaliador determine a obrigatoriedade de determinada frase na pesquisa. Entretanto, assim como vimos anteriormente, a utilização do predicado CONTAINS não retorna o SCORE.

Da mesma maneira que é possível integrar o predicado CONTAINS à consulta, também podem ser integrados outros operadores como NEAR, termo de proximidade, e o NOT, para determinar quais palavras não devem aparecer nas respostas, que permitem ao avaliador uma filtragem maior dos resultados retornados. Como podemos observar na Figura 29, além da presença da palavra “Sudeste” ser um dos pré-requisitos para que a resposta seja retornada, é necessário que a frase “Ventos Alísios” também esteja presente. Com a função CONTAINS, apenas as respostas que tiverem a frase “Ventos alísios” serão retornadas, porém, não é possível obter o SCORE da frase.

Figura 29. Consulta as respostas que possuem a palavra “Sudeste” e a frase “Ventos alísios” no SQL Server.

```
SELECT s.keyphrase, s.SCORE, d.DocId, d.Questão, d.Resposta
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
INNER JOIN BD_Resposta AS d
ON s.document_key = d.DocId
WHERE s.keyphrase = 'Sudeste' and CONTAINS(Resposta, '"Ventos alísios"')
and DocId > 1
ORDER BY s.SCORE DESC
```

Fonte: Desenvolvido pelo autor

A Figura 30 apresenta os resultados obtidos na consulta da Figura 29. Nesse exemplo, as respostas correta e meio certa foram retornadas, sendo que ambas possuem a palavra “Sudeste” e a frase “Ventos alísios”.

Figura 30. Resultados obtidos na consulta da Figura 29.

	keyphrase	score	DocId	Questão	Resposta
1	sudeste	1	4	Resposta correta	A direção dos ventos alísios no litoral brasilei...
2	sudeste	1	5	Resposta meio certa	A direção dos ventos alísios no litoral brasilei...

Fonte: Desenvolvido pelo autor

O predicado CONTAINS exige que a(s) resposta(s) contenha(m) exatamente o mesmo conteúdo especificado. Como alternativa, temos o predicado FREETEXT, que permite a expansão da pesquisa. Se substituirmos o predicado CONTAINS pelo FREETEXT (Figura 31), poderemos ampliar a busca ao efetuar pesquisas por respostas relacionadas ao termo/frase especificado, sem a obrigatoriedade do(a) termo/frase ser exatamente igual ao especificado. Na Figura 32, as respostas relacionadas a “Ventos alísios vindo do sul” são buscadas e obtemos a “Resposta correta” e “Resposta meio certa”, mesmo que em nenhuma das respostas exista uma frase exatamente igual à “Ventos alísios vindo do sul”. A frase, que fica à critério do avaliador, pode ser usada para identificar as respostas que possuem relação com a resposta que se julga certa. Diferente do uso de CONTAINS, quando o FREETEXT é utilizado, não é necessária a utilização de aspas duplas.

Figura 31. Consulta as respostas que possuem a palavra “Sudeste” e que possuem relação com a frase “Ventos alísios vindo do sul” no SQL Server.

```
SELECT  s.keyphrase, s.SCORE, d.DocId, d.Questão, d.Resposta
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
      INNER JOIN BD_Resposta AS d
      ON s.document_key = d.DocId
WHERE s.keyphrase = 'Sudeste' and FREETEXT(Resposta, 'Ventos alísios
vindo do sul') and DocId > 1
ORDER BY s.SCORE DESC
```

Fonte: Desenvolvido pelo autor

Figura 32. Resultados obtidos na consulta da Figura 31.

	keyphrase	score	DocId	Questão	Resposta
1	sudeste	1	4	Resposta correta	A direção dos ventos alísios no litoral brasilei...
2	sudeste	1	5	Resposta meio certa	A direção dos ventos alísios no litoral brasilei...

Fonte: Desenvolvido pelo autor

D) OBTÉM A QUANTIDADE DE PALAVRAS-CHAVE DETERMINADAS PELO AVALIADOR PRESENTES EM CADA RESPOSTA

Caso o avaliador queira verificar a quantidade de palavras relevantes para ele em cada resposta, ele poderá utilizar a função COUNT, como na consulta apresentada na Figura 33.

Figura 33. Consulta a quantidade de palavras que foram especificadas pelo avaliador e que estão presentes em cada resposta.

```
SELECT d.Questão, COUNT(s.keyphrase) 'Qtd de palavras relevantes'
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
    INNER JOIN BD_Resposta AS d
    ON s.document_key = d.DocId
WHERE s.keyphrase in ('Sudeste', 'Noroeste', 'Ventos') and d.DocId > 1
GROUP BY d.Questão
ORDER BY COUNT(s.keyphrase)
```

Fonte: Desenvolvido pelo autor

Na Figura 34 são apresentados os resultados obtidos na consulta da Figura 33, e podemos perceber que a “Resposta certa” e “Resposta meio certa”, possuem 3 palavras chaves definidas pelo avaliador, enquanto a “Resposta incompleta”, apresenta apenas 1 das palavras chaves indicadas. É importante salientar, que essa contagem de palavras é feita somente uma vez, mesmo que a palavra esteja repetida.

Figura 34. Resultados obtidos na consulta (Figura 33).

	Questão	Qtd de palavras relevantes
1	Resposta incompleta	1
2	Resposta meio certa	3
3	Resposta correta	3

Fonte: Desenvolvido pelo autor

4. 2. SEGUNDA POSSIBILIDADE DE ANÁLISE: RESPOSTAS QUE O SQL SERVER COMPARA AUTOMATICAMENTE SEM A NECESSIDADE DE ESPECIFICAR PALAVRAS-CHAVE

A) OBTÉM AS PALAVRAS MAIS RELEVANTES DE TODAS AS RESPOSTAS

O avaliador também consegue ter acesso às palavras de todas as respostas a partir da função SEMANTICKEYPHRASETABLE. A consulta apresentada na Figura 35 retorna todas as respostas que possuem palavras com SCORE superior à 0.6.

Figura 35. Consulta que obtém as palavras com SCORE superior à 0.6 em todas as respostas.

```
SELECT keyphrase, Questão, SCORE
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
INNER JOIN BD_Resposta AS d
ON s.document_key = d.DocId
WHERE d.DocId>1 and s.SCORE > 0.6
ORDER BY keyphrase, s.SCORE DESC
```

Fonte: Desenvolvido pelo autor

A Figura 36 apresenta os resultados obtidos na consulta da Figura 35, onde é possível observar a distribuição das palavras e dos SCORES para cada resposta.

Figura 36. Resultados obtidos na consulta da Figura 35.

	keyphrase	Questão	score
1	áfrica	Resposta correta	1
2	áfrica	Resposta meio certa	1
3	brasil	Resposta meio certa	0,9535604
4	brasil	Resposta correta	0,9535604
5	direção	Resposta incompleta	1
6	noroeste	Resposta correta	1
7	noroeste	Resposta meio certa	1
8	sudeste	Resposta meio certa	1
9	sudeste	Resposta correta	1

Fonte: Desenvolvido pelo autor

B) OBTÉM AS PALAVRAS MAIS RELEVANTES DE UMA RESPOSTA

Também é possível especificar uma resposta que o avaliador deseja analisar especificamente, de modo a obter o SCORE de suas palavras. Assim como vimos anteriormente, é preciso apenas o uso de um terceiro parâmetro na função SEMANTICKEYPHRASETABLE referente ao identificador da resposta desejada, como o apresentado na consulta da Figura 37. Neste exemplo, para que as palavras sejam retornadas, é necessário que elas tenham obtido um SCORE superior a 0.7.

Figura 37. Consulta que obtém as palavras contidas na resposta do professor (DocId = 1) que possuem SCORE superior à 0.7.

```
SELECT s.keyphrase, Questão, SCORE
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta, 1) AS s
INNER JOIN BD_Resposta AS d
ON s.document_key = d.DocId
WHERE s.SCORE > 0.7
ORDER BY s.SCORE DESC
```

Fonte: Desenvolvido pelo autor

A Figura 38 apresenta os resultados obtidos na consulta da Figura 37. Nela podemos ver que as palavras “litoral”, “navegação”, “noroeste”, “sudeste” e “ventos”, obtiveram SCORE igual à 1 na “Resposta do professor”.

Figura 38. Resultados obtidos na consulta da Figura 37.

	keyphrase	Questão	score
1	litoral	Resposta do Professor	1
2	navegação	Resposta do Professor	1
3	noroeste	Resposta do Professor	1
4	sudeste	Resposta do Professor	1
5	ventos	Resposta do Professor	1

Fonte: Desenvolvido pelo autor

C) COMPARA A RESPOSTA DO PROFESSOR COM AS DEMAIS RESPOSTAS E OBTÉM O SCORE

Outra função de pesquisa semântica que o SQL Server apresenta é a SEMANTICSIMILARITYTABLE, que possibilita ao avaliador comparar uma resposta específica com as demais respostas, de modo a identificar a semelhança (SCORE). No exemplo de consulta da Figura 39, a resposta do professor (DocId = 1) é comparada com as demais respostas.

Figura 39. Consulta que obtém o grau de similaridade de cada resposta em relação a resposta do professor (DocId = 1).

```

SELECT      s.matched_document_key, s.SCORE, d.Questão
FROM SEMANTICSIMILARITYTABLE(BD_Resposta, Resposta, 1) AS s
      INNER JOIN BD_Resposta AS d
      ON s.matched_document_key = d.DocId
ORDER BY s.SCORE DESC

```

Fonte: Desenvolvido pelo autor

Como podemos ver na Figura 40, a “Resposta incorreta” e a “Resposta incompleta”, apresentaram menor SCORE em relação a resposta do professor. Enquanto isso, a resposta “Resposta correta” e a “Resposta meio certa”, não só apresentaram maior SCORE, como apresentaram o mesmo SCORE. É importante lembrar, que o que diferencia essas respostas é a inversão das direções dos ventos, tendo estas as mesmas palavras. Por conta disso, era esperado que o SCORE de ambas fosse o mesmo, já que o SQL Server não tem a capacidade de determinar se uma resposta está certa ou errada, mas sim disponibilizar informações que podem ser úteis ao avaliador.

Figura 40. Resultados obtidos na consulta da Figura 39.

	matched_document_key	score	Questão
1	4	0,6420966	Resposta correta
2	5	0,6420966	Resposta meio certa
3	3	0,1276662	Resposta incompleta
4	2	0,0531088	Resposta incorreta

Fonte: Desenvolvido pelo autor

Para contornar situações como essas, em que uma inversão de conceitos passou despercebida, outros operadores podem ser usados na tentativa de filtrar as respostas dos alunos, como o CONTAINS e o NEAR. Para exemplificar, temos uma consulta na Figura 41, na qual o operador NEAR é utilizado de modo a fazer com que a palavra “Brasil” esteja próxima da frase “do sudeste” (nesta ordem) e que a palavra “África” esteja próxima da frase “do noroeste”, em todas as respostas retornadas.

Figura 41. Consulta que obtém o grau de similaridade das respostas em relação à resposta do professor que possuem SCORE superior à 0.6 e que contém a palavra “Brasil” perto da frase “do sudeste” e a palavra “África” perto da frase “do noroeste”.

```

SELECT      s.matched_document_key, s.SCORE, d.Questão
FROM SEMANTICSIMILARITYTABLE(BD_Resposta, Resposta, 1) AS
      INNER JOIN BD_Resposta AS d
      ON s.matched_document_key = d.DocId
WHERE  SCORE > 0.6 and
      CONTAINS(Resposta, 'NEAR(("brasil", "do sudeste"), 2, TRUE)') and
      CONTAINS(Resposta, 'NEAR(("áfrica", "do noroeste"), 2, TRUE)')
ORDER BY s.SCORE DESC

```

Fonte: Desenvolvido pelo autor

Como pode ser observado na Figura 42, com o uso do operador NEAR (Figura 41), apenas a “Resposta correta” foi retorna, uma vez que apenas essa resposta atendia a todas as especificações indicadas pelo avaliador (palavra “Brasil” perto de sudeste e palavra “África” perto de noroeste). Esse tipo de filtragem é útil quando o avaliador já tem consciência de possíveis inversões de conceitos que podem ocorrer nas respostas.

Figura 42. Resultados obtidos na consulta da Figura 41.

	matched_document_key	score	Questão
1	4	0,6420966	Resposta correta

Fonte: Desenvolvido pelo autor

Também é possível obter as respostas que possuem as formas flexionadas de uma palavra específica, usando FORMSOF com o parâmetro INFLECTIONAL. Por exemplo, para verbos é possível considerar as conjugações e, para substantivos, as formas singular, plural e possessiva. A Figura 43 apresenta uma consulta que retorna as respostas que possuem as formas flexionadas da palavra “contribuir”.

Figura 43. Consulta que obtém as respostas que possuem o grau de similaridade superior à 0.6 em relação à resposta do professor (DocId = 1) e que possuem a forma flexionada da palavra “contribuir”.

```
SELECT      s.matched_document_key, s.SCORE, d.Questão
FROM SEMANTICSIMILARITYTABLE(BD_Resposta, Resposta, 1) AS s
      INNER JOIN BD_Resposta AS d
      ON s.matched_document_key = d.DocId
WHERE SCORE > 0.6 and
      CONTAINS(Resposta,'FORMSOF(INFLECTIONAL,contribuir) ')
ORDER BY s.SCORE DESC
```

Fonte: Desenvolvido pelo autor

Conforme apresentado na Figura 44, foram retornadas a “Resposta correta” e a “Resposta meio certa”, pois ambas possuem uma das conjugações do verbo “contribuir”, que é “contribua”.

Figura 44. Resultados obtidos na consulta da Figura 43.

matched_document_key	score	Questão
4	0,7917174	Resposta correta
5	0,7917174	Resposta meio certa

Fonte: Desenvolvido pelo autor

Podemos usar também FORMSOF com o parâmetro THESAURUS, de modo a expandir a pesquisa e encontrar as respostas que possuem os sinônimos de uma palavra especificada, de acordo com um dicionário de sinônimos. No exemplo de consulta da Figura 45, obtemos as respostas que possuem sinônimos da palavra ‘auxilie’, definido no dicionário de sinônimos apresentado na Figura 47.

Figura 45. Consulta que obtém as respostas que possuem o grau de similaridade superior à 0.6 em relação à reposta do professor (DocId = 1) e que contêm alguns dos sinônimos da palavra “auxilie”.

```

SELECT      s.matched_document_key, s.SCORE, d.Questão
FROM SEMANTICSIMILARITYTABLE(BD_Resposta, Resposta, 1) AS s
      INNER JOIN BD_Resposta AS d
      ON s.matched_document_key = d.DocId
WHERE SCORE > 0.6 and
      CONTAINS(Resposta,'FORMSOF(THESAURUS,auxilie) ')
ORDER BY s.SCORE DESC

```

Fonte: Desenvolvido pelo autor

De acordo com os resultados apresentados na Figura 46, foram retornadas a “Resposta correta” e a “Resposta meio certa”, pois ambas possuem um dos sinônimos definidos para o verbo “auxilie”, que é “contribua”.

Figura 46. Resultados obtidos na consulta da Figura 45.

matched_document_key	score	Questão
4	0,7917174	Resposta correta
5	0,7917174	Resposta meio certa

Fonte: Desenvolvido pelo autor

A Figura 47 apresenta o dicionário de sinônimos que foi desenvolvido pelo autor com o intuito de exemplificar o funcionamento da consulta da Figura 45. Esse dicionário foi especificado como um arquivo XML e contém as palavras sinônimas “contribua”, “auxilie”, “coopere”, “ajude” e “facilite”.

Figura 47. Dicionário de sinônimos.

```
<XML ID="Microsoft Search Thesaurus">
  <thesaurus xmlns="x-schema:tsSchema.xml">
    <diacritics_sensitive>0</diacritics_sensitive>
    <expansion>
      <sub> contribua </sub>
      <sub> auxilie </sub>
      <sub> coopere </sub>
      <sub> ajude </sub>
      <sub> facilite </sub>
    </expansion>
  </thesaurus>
</XML>
```

Fonte: Desenvolvido pelo autor

O SQL Server oferece um conjunto de arquivos XML de dicionários de sinônimos para cada idioma. A princípio, esses arquivos são praticamente vazios. Para carregar um arquivo de dicionário de sinônimos, basta apenas executar 'sp_fulltext_load_thesaurus_file 1046' (isso para o idioma Português - Brasil). O *script* que mostra todo o processo de execução e criação do dicionário de sinônimos está disponível no APENDICE A deste trabalho.

D) OBTÉM AS PALAVRAS EM COMUM EM DUAS RESPOSTAS

A função de pesquisa semântica SEMANTICSIMILARITYDETAILSTABLE possibilita comparar duas respostas de modo a identificar palavras que possuem em comum, bem como os seus respectivos SCORE. A Figura 48 apresenta uma consulta que compara as respostas correta e meio certa, obtendo as palavras que ambas têm em comum e cujo SCORE seja maior que 0.6. O resultado foram as palavras “África”, “noroeste” e “sudeste”, que obtiveram SCORE máximo, e “Brasil” que obteve SCORE de aproximadamente 0,9 (Figura 49).

Figura 48. Consulta que obtém as palavras que a resposta correta (DocId = 4) e a resposta meio certa (DocId = 5) possuem em comum, com SCORE superior à 0.6.

```
SELECT s.keyphrase, s.SCORE
FROM SEMANTICSIMILARITYDETAILSTABLE(BD_Resposta, Resposta, 4,
Resposta, 5) AS s
WHERE S.SCORE > 0.6
ORDER BY s.SCORE DESC
```

Fonte: Desenvolvido pelo autor

Figura 49. Resultados obtidos na consulta da Figura 48.

	keyphrase	score
1	áfrica	1
2	noroeste	1
3	sudeste	1
4	brasil	0,9092774

Fonte: Desenvolvido pelo autor

E) OBTÉM A QUANTIDADE DE PALAVRAS EM COMUM EM DUAS RESPOSTAS

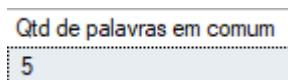
Usando a função COUNT, é possível obter a quantidade de palavras que possuem um determinado SCORE (definido pelo avaliador). A consulta da Figura 50 compara a resposta do professor com a resposta meio certa, obtendo a quantidade de palavras que ambas possuem em comum e com SCORE superior à 0.3. O resultado foram 5 palavras, como mostrado a Figura 51.

Figura 50. Consulta que obtém a quantidade de palavras que a respostas do professor (DocId = 1) e a resposta meio certa (DocId = 5) possuem em comum, com SCORE superior à 0.3.

```
SELECT COUNT(s.keyphrase) 'Qtd de palavras em comum'
FROM SEMANTICSIMILARITYDETAILSTABLE(BD_Resposta, Resposta, 1,
Resposta, 5) AS s
WHERE s.SCORE > 0.3
ORDER BY COUNT(s.keyphrase)
```

Fonte: Desenvolvido pelo autor

Figura 51. Resultados obtidos na consulta da Figura 50.



Qtd de palavras em comum
5

Fonte: Desenvolvido pelo autor

Este estudo de caso utilizou apenas quatro respostas a serem avaliadas em relação à resposta do professor, de modo a apresentar as possibilidades de análise. Entretanto, considerando um número maior de respostas, é possível ainda incrementar as consultas de modo a facilitar a análise dos resultados pelo avaliador, por meio da utilização de alguns filtros:

- O avaliador pode determinar o valor exato ou o intervalo da medida de relevância. Por exemplo, obter apenas as respostas que possuem SCORE igual à 0.3, usando o operador de igualdade (=); obter as respostas que possuem SCORE superior ou inferior à 0.3, por meio dos operadores de comparação > (maior) e < (menor), respectivamente; obter as respostas com SCORE entre dois valores, utilizando a cláusula BETWEEN;
- O avaliador pode indicar o número de linhas que deseja retornar em uma consulta, usando a cláusula TOP. Por exemplo, 'TOP(3) keyphrase' retorna as três palavras com maior SCORE de todo o documento.

Esses filtros possibilitam ao avaliador agrupar as respostas que possuem similaridades entre si e analisar cada grupo de respostas de uma maneira específica, como, por exemplo, agrupar as melhores respostas (acima de um determinado valor de SCORE) para iniciar a avaliação.

5. CONSIDERAÇÕES FINAIS

O presente trabalho, abordou as diversas funcionalidades oferecidas pelos Sistemas de Gerenciamento de Banco de Dados (SGBD) em relação aos tipos de dados *Full Text*, optando em estudar mais profundamente a capacidade do SQL Server na análise de questões discursivas por meio da Pesquisa Semântica.

Após a construção de um banco de dados de repostas com peculiaridades bem definidas (certa, meio certa, incompleta e incorreta), foi possível identificar as possibilidades de análise das respostas por meio dos recursos oferecidos pelo SQL server.

Apesar do SGBD em questão não permitir a pesquisa semântica por frases, obtivemos resultados bastante satisfatórios, pois existem recursos que podem ser utilizados em conjunto de modo a proporcionar várias tipos de análises, sendo possível: classificar as respostas a partir da ponderação dos termos; obter a quantidade de palavras definidas pelo avaliador presentes nas respostas; obter o valor da similaridade das respostas em relação à resposta correta; e ampliar o escopo da pesquisa por palavras a partir do uso de dicionário de sinônimos.

Nesse sentido, a utilização de recursos de um gerenciador de banco de dados, como o SQL Server, como apoio à avaliação de questões discursivas apresentou inúmeros benefícios ao avaliador, que poderá realizar seu trabalho de maneira mais rápida e eficiente, visto que poderá obter informações relevantes para a sua avaliação e também agrupar as respostas de modo a facilitar o processo de correção.

Para trabalhos futuros, sugere-se o desenvolvimento do estudo de caso utilizando os outros dois SGBD estudados (seção 3.2), o Oracle e o DB2, de modo a avaliar também as suas funcionalidades em relação a análise de questões discursivas, bem como aplicar as análises propostas considerando um banco de dados com um número maior de respostas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ARCANECODE. **Creating custom thesaurus entries in Sql Server 2005 and 2008 Full Text Search.** 2008. Disponível em:

<<https://arcanecode.com/2008/05/28/creating-custom-thesaurus-entries-in-sql-server-2005-and-2008-full-text-search/>>. Acesso em: 19 de Novembro de 2020.

ARCANECODE. **Using FORMSOF in Sql Server Full Text Searching.** 2008. Disponível em: <<https://arcanecode.com/2008/05/27/using-formsof-in-sql-server-full-text-searching/>>. Acesso em: 19 de Novembro de 2020.

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: 19 de Novembro de 2020.

BERASATEGI, Ane. **Semantic Search.** 2019. Disponível em: <<https://towardsdatascience.com/semantic-search-73fa1177548f>>. Acesso em: 18 de Março de 2020.

DBA PRESENTS. **Full-Text Search - practical introduction.** 2014. Disponível em: <<https://dba-presents.com/index.php/component/content/article?id=21:full-text-search-practical-introduction>>. Acesso em: 20 de Outubro de 2020.

DEVMEDIA. **Full Text Search no Oracle.** 2008. Disponível em: <<https://www.devmedia.com.br/full-text-search-no-oracle-parte-01/7793>>. Acesso em: 30 de Março de 2020.

DEVMEDIA. **Implementação de índices Textuais (Full-Text Search) no IBM DB2.** 2009. Disponível em: <<https://www.devmedia.com.br/implementacao-de-indices-textuais-full-text-search-no-ibm-db2/14166>>. Acesso em: 16 de Julho de 2020.

DEVMEDIA. **Implementação de Índices Textuais (Full-Text Search) no IBM DB2.** S/D. Disponível em: <<https://www.devmedia.com.br/implementacao-de-indices-textuais-full-text-search-no-ibm-db2/14166>>. Acesso em: 24 de Dezembro de 2020.

EISENBERG, Andrew; MELTON, Jim. **SQL Multimedia and Application Packages (SQL/MM).** 2001. Disponível em: <

<https://sigmodrecord.org/publications/sigmodRecord/0112/standards.pdf>> Acesso em: 16 de Julho de 2020.

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: 16 de Julho de 2020.

FORD, Roger. **Oracle Text In Oracle Database 12c**. 2017. Disponível em: <<https://www.oracle.com/technetwork/database/12coraclexttwp-1961244.pdf>> Acesso em: 16 de Julho de 2020.

GARCIA, Gabriela. **Descubra os melhores métodos de avaliação para seus alunos**. 2017. Disponível em: <<https://www.provafacilnaweb.com.br/blog/metodos-de-avaliacao/>>. Acesso em: 13 de Dezembro de 2020.

IBM. **Attribute data types**. S/D. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SSLKT6_7.6.0/com.ibm.mt.doc/configur/r_attribute_data_types.html >. Acesso em: 16 de Julho de 2020.

IBM. **Basic Search**. S/D. Disponível em: <https://www.ibm.com/support/producthub/db2/docs/content/SSEPGG_11.5.0/com.ibm.db2.luw.admin.ts.doc/doc/t0058802.html>. Acesso em: 20 de Julho de 2020.

IBM. **Db2 Text Search and NSE semantic and syntactic differences for searching**. S/D. Disponível em: <https://www.ibm.com/support/producthub/db2/docs/content/SSEPGG_11.5.0/com.ibm.db2.luw.admin.ts.doc/doc/c0059118.html >. Acesso em: 20 de Julho de 2020.

IBM. **DB2 Text Search key features and concepts**. DB2 10.5 for Linux, Unix, and Windows. S/D. Disponível em: <https://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.ts.doc/doc/c0057450.html> Acesso em: 16 de Julho de 2020.

IBM. **Db2 Text Search.** S/D. Disponível em:
<https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.db2.luw.admin.ts.doc/doc/c0051296.html>. Acesso em: 16 de Julho de 2020.

IBM. **Full text search.** 2008. Disponível em:
<<https://developer.ibm.com/technologies/databases/tutorials/dm-0810shettar/#search-with-db2-text-search>>. Acesso em: 05 de Junho de 2020.

IBM. **Scenario: Indexing and searching.** DB2 11.1. S/D. Disponível em:
<https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.db2.luw.admin.ts.doc/doc/c0053088.html> Acesso em: 16 de Julho de 2020.

ISO/IEC 13249-2 Information technology. **SQL Multimedia and Application Packages - Part 2: Full-Text.** International Organization For Standardization, 2003.

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

MICROSOFT. **Configurar e gerenciar arquivos de dicionário de sinônimos para Pesquisa de texto completo.** 2020. 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#:~:text=SQL%20Server%20oferece%20um%20conjunto,para%20cada%20idioma%20com%20suporte.&text=Eles%20cont%C3%AAm%20apenas%20a%20estrutura,de%20sin%C3%B4nimos%20de%20exemplo%20comentado>>. Acesso em: 19 de Novembro de 2020.

MICROSOFT. **Configurar e gerenciar filtros para pesquisa.** 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/search/configure-and-manage-filters-for-search?view=sql-server-ver15>>. Acesso em: 16 de Julho de 2020.

MICROSOFT. **Consultar dados espaciais de vizinho mais próximo.** 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/spatial/query-spatial-data-for-nearest-neighbor?view=sql-server-ver15>>. Acesso em: 11 de Dezembro de 2020.

MICROSOFT. **COUNT (Transact-SQL)**. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/t-sql/functions/count-transact-sql?view=sql-server-ver15>>. Acesso em: 19 de Agosto de 2020.

MICROSOFT. **Localizar documentos semelhantes e relacionados com a pesquisa semântica**. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/search/find-similar-and-related-documents-with-semantic-search?view=sql-server-ver15>>. Acesso em: 20 de Agosto de 2020.

MICROSOFT. **Localizar frases chave em documentos com pesquisa semântica**. 2017. 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: 16 de Julho de 2020.

MICROSOFT. **Pesquisa de Texto Completo**. 2018. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/search/full-text-search?view=sql-server-ver15>>. Acesso em: 18 de Março de 2020.

MICROSOFT. **Pesquisa de Texto Completo**. 2018. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/search/full-text-search?view=sql-server-ver15>>. Acesso em: 20 de Agosto de 2020.

MICROSOFT. **Pesquisa Semântica (SQL Server)**. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/search/semantic-search-sql-server?view=sql-server-ver15>>. Acesso em: 30 de Março de 2020.

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: 20 de Agosto de 2020.

MICROSOFT. **Semantickeyphrasetable**. 2016. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/system-functions/semantickeyphrasetable-transact-sql?view=sql-server-ver15>>. Acesso em: 20 de Agosto de 2020.

MICROSOFT. **Semanticsimilaritydetailstable**. 2016. 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: 20 de Agosto de 2020.

MICROSOFT. **Semanticsimilaritytable**. 2016. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/system-functions/semanticsimilaritytable-transact-sql?view=sql-server-ver15>>. Acesso em: 20 de Agosto de 2020.

MICROSOFT. **Versões de idiomas locais no SQL Server**. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/sql-server/install/local-language-versions-in-sql-server?view=sql-server-ver15>>. Acesso em: 20 de Agosto de 2020.

MICROSOFT. **WHERE (Transact-SQL)**. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/t-sql/queries/where-transact-sql?view=sql-server-ver15>>. Acesso em: 11 de Dezembro de 2020.

MYSQL. **Full-Text Search Functions**. S/D. Disponível em: <<https://dev.mysql.com/doc/refman/8.0/en/fulltext-search.html>>. Acesso em: 16 de Julho de 2020.

ORACLE. **Application Developer's Guide: Working With a Thesaurus in Oracle Text**. S/D. Disponível em: <<https://docs.oracle.com/en/database/oracle/oracle-database/12.2/ccapp/working-with-thesaurus-in-oracle-text.html#GUID-C17963C5-1201-4243-952A-91C0C9EE4B15>>. Acesso em: 24 de Dezembro de 2020.

ORACLE. **Database Semantic Technologies Developer's Guide**. S/D. Disponível em: <https://docs.oracle.com/cd/E11882_01/appdev.112/e25609/indexing_for_docs.htm#RDFRM99921>. Acesso em: 28 de Junho de 2020.

ORACLE. **Text Application Developer's Guide: Oracle Text Thesaurus Creation and Maintenance**. S/D. Disponível em:

<<https://docs.oracle.com/database/121/CCAPP/GUID-C1DB8E0C-B8F7-4833-8257-B0FD3B57636A.htm#CCAPP9326>>. Acesso em: 24 de Dezembro de 2020.

PAULO, Vicente. **Questões Objetivas X Questões Discursivas**. 2013. Disponível em: <<https://www.pontodosconcursos.com.br/artigo/9380/vicente-paulo/questoes-objetivas-x-questoes-discursivas>>. Acesso em: 13 de Dezembro de 2020.

POSTGRESQL. **Full Text Search**. 2018. Disponível em: <<https://www.postgresql.org/docs/12/textsearch.html>>. Acesso em: 22 de Junho de 2020.

SUPPORT MOZILLA. **Abreviação de Localização**. S/D. Disponível em: <<https://support.mozilla.org/pt-BR/kb/abreviacao-de-localizacao>>. Acesso em: 19 de Novembro de 2020.

APÊNDICE A – *Script* da aplicação do estudo de caso desenvolvido no SQL Server

-- Cria o banco de dados

```
CREATE DATABASE BD_Resposta
```

```
USE BD_Resposta
```

```
GO
```

-- Cria tabela de documentos

```
CREATE TABLE BD_Resposta
```

```
(
```

```
    DocId INT IDENTITY(1,1) NOT NULL,
```

```
    Questão VARCHAR(50) NOT NULL,
```

```
    Resposta VARCHAR(500) NOT NULL,
```

```
    CONSTRAINT PK_BD_Resposta PRIMARY KEY CLUSTERED(DocId)
```

```
);
```

```
GO
```

-- Verifica se fulltext está habilitado

```
SELECT    CASE
```

```
            WHEN SERVERPROPERTY('ISFULLTEXTINSTALLED') = 1 then
```

```
'FULLTEXT HABILITADO'
```

```
            ELSE 'FULLTEXT DESABILITADO'
```

```
            END AS 'FullText'
```

```
GO
```

-- Insere os dados na tabela

```
INSERT INTO BD_Resposta(Questão, Resposta)
```

```
VALUES ('Resposta do Professor', 'A direção dos ventos alísios no litoral brasileiro é de sudeste para noroeste; no litoral africano, é de noroeste para sudeste. O desenvolvimento das técnicas de navegação naquele contexto estava relacionado ao aproveitamento do regime dos ventos, dados que as embarcações eram movidas a vela. Nesse sentido, a navegação da África para o Brasil era facilitada. ')
```

GO

```
INSERT INTO BD_Resposta(Questão, Resposta)
```

```
VALUES ('Resposta incorreta', 'O comércio de pessoas escravizadas desempenhou papel fundamental na dinamização de vários tipos de trocas estabelecidas entre o Brasil e África no contexto da colonização. Por meio de uma multiplicidade de relações potencializadas pelo tráfico, formou-se um amplo complexo histórico e cultural afro-brasileiro. ')
```

GO

```
INSERT INTO BD_Resposta(Questão, Resposta)
```

```
VALUES ('Resposta incompleta', 'Os ventos alísios, que sopram no Hemisfério Sul na proximidade do Trópico de Capricórnio em direção ao Equador, sofrem inclinação devido à rotação da Terra (Efeito Coriolis) e adquirem direção aproximada oeste-leste atuando durante o ano todo. Como antigamente as embarcações eram movidas à vela, os ventos facilitavam e permitiam que o transporte marítimo ocorresse. ')
```

GO

```
INSERT INTO BD_Resposta(Questão, Resposta)
```

```
VALUES ('Resposta correta', 'A direção dos ventos alísios no litoral brasileiro são do sudeste para o noroeste, enquanto isso, no litoral africano são do noroeste para o sudeste. Como antigamente as embarcações eram movidas à vela, a direção dos ventos era de extrema importância para a navegação. Dessa forma, a direção dos ventos contribua para que o transporte marítimo da África para o Brasil fosse facilitado. ')
```

GO

```
INSERT INTO BD_Resposta(Questão, Resposta)
```

```
VALUES ('Resposta meio certa', 'A direção dos ventos alísios no litoral brasileiro são do noroeste para o sudeste, enquanto isso, no litoral africano são do sudeste para o noroeste. Como antigamente as embarcações eram movidas à vela, a direção dos ventos era de extrema importância para a navegação. Dessa forma, a direção dos
```

ventos contribua para que o transporte marítimo da África para o Brasil fosse facilitado. ')

GO

-- Verifica conteúdo da tabela de documentos

SELECT * FROM BD_Resposta

GO

-- Carregar os resources

-- Após a carga dos resources reiniciar o SQL Server

EXEC sys.sp_fulltext_service 'load_os_resources', 1;

GO

-- Verifica os componentes para filtro

EXEC sys.sp_help_fulltext_system_components 'filter';

GO

-- Lista de linguagens disponíveis no fulltext

SELECT lcid, name

FROM sys.fulltext_languages

ORDER BY 2;

GO

-- Instalar SemanticLanguageDatabase.msi -- <https://codingsight.com/how-to-use-semantic-search-in-sql-server/>

-- Criar a database semanticssdb

CREATE DATABASE semanticssdbBDResposta ON

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

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

FOR ATTACH;

GO

```

-- Registrar a database semanticsdb
EXEC sp_fulltext_semantic_register_language_statistics_db @dbname =
N'semanticsdbBDResposta';
GO

-- Criar catalogo fulltext
CREATE FULLTEXT CATALOG catalogofulltext;
GO

-- Criando índices
CREATE UNIQUE INDEX indice ON BD_Resposta(DocId);

CREATE FULLTEXT INDEX ON BD_Resposta
(
    Resposta          -- coluna que será indexada
    Language 1046 STATISTICAL_SEMANTICS    -- 1046 - LCID Brasil - portugues
)
KEY INDEX indice ON catalogofulltext --Unique index
WITH CHANGE_TRACKING AUTO          -- Population type;
GO

-- Consultas realizadas no estudo de caso
SELECT FT_TBL.Questão, KEY_TBL.RANK, FT_TBL.Resposta
FROM BD_Resposta AS FT_TBL
    INNER JOIN CONTAINSTABLE(BD_Resposta, Resposta,
        'ISABOUT ("Sudeste" WEIGHT (.1),
        "Noroeste" WEIGHT (.1), "Ventos alísios" WEIGHT (.1), "Navegação" WEIGHT
        (.2))' ) AS KEY_TBL
    ON FT_TBL.DocId = KEY_TBL.[KEY]
WHERE FT_TBL.DocId > 1 -- retirando a resposta do professor
ORDER BY KEY_TBL.RANK DESC;
GO

SELECT s.keyphrase, s.SCORE, d.DocId, d.Questão, d.Resposta

```

```

FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
    INNER JOIN BD_Resposta AS d
        ON s.document_key = d.DocId
WHERE s.keyphrase in ('Sudeste', 'Noroeste', 'Ventos' ) and d.DocId > 1
ORDER BY s.SCORE DESC;
GO

```

```

SELECT  s.keyphrase, s.SCORE, d.Questão, d.Resposta
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta, 4) AS s
    INNER JOIN BD_Resposta AS d
        ON s.document_key = d.DocId
WHERE s.keyphrase in ('Sudeste', 'Noroeste', 'Ventos')
ORDER BY s.SCORE DESC;
GO

```

```

SELECT  s.keyphrase, s.SCORE, d.DocId, d.Questão, d.Resposta
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
    INNER JOIN BD_Resposta AS d
        ON s.document_key = d.DocId
WHERE s.keyphrase = 'Sudeste' and CONTAINS(Resposta, "'Ventos alísios'") and
DocId > 1
ORDER BY s.SCORE DESC;
GO

```

```

SELECT  s.keyphrase, s.SCORE, d.DocId, d.Questão, d.Resposta
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
    INNER JOIN BD_Resposta AS d
        ON s.document_key = d.DocId
WHERE s.keyphrase = 'Sudeste' and FREETEXT(Resposta, 'Ventos alísios vindo do
sul') and DocId > 1
ORDER BY s.SCORE DESC;
GO

```

```

SELECT d.Questão, COUNT(s.keyphrase) 'Qtd de palavras relevantes'

```

```

FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
    INNER JOIN BD_Resposta AS d
        ON s.document_key = d.DocId
WHERE s.keyphrase in ('Sudeste', 'Noroeste', 'Ventos') and d.DocId > 1
GROUP BY d.Questão
ORDER BY COUNT(s.keyphrase);
GO

```

```

SELECT keyphrase, Questão, SCORE
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta) AS s
    INNER JOIN BD_Resposta AS d
        ON s.document_key = d.DocId
WHERE d.DocId>1 and s.SCORE > 0.6
ORDER BY keyphrase, s.SCORE DESC;
GO

```

```

SELECT s.keyphrase, Questão, SCORE
FROM SEMANTICKEYPHRASETABLE (BD_Resposta, Resposta, 1) AS s
    INNER JOIN BD_Resposta AS d
        ON s.document_key = d.DocId
WHERE s.SCORE > 0.7
ORDER BY s.SCORE DESC
GO

```

```

SELECT      s.matched_document_key, s.SCORE, d.Questão
FROM SEMANTICSIMILARITYTABLE(BD_Resposta, Resposta, 1) AS s -- em
comparação a resposta do professor
    INNER JOIN BD_Resposta AS d
        ON s.matched_document_key = d.DocId
ORDER BY s.SCORE DESC;
GO

```

```

SELECT      s.matched_document_key, s.SCORE, d.Questão
FROM SEMANTICSIMILARITYTABLE(BD_Resposta, Resposta, 1) AS s

```

```

        INNER JOIN BD_Resposta AS d
        ON s.matched_document_key = d.DocId
WHERE SCORE > 0.6 and CONTAINS(Resposta, 'NEAR(("brasil", "do sudeste"),
2, TRUE)') and CONTAINS(Resposta, 'NEAR(("áfrica", "do noroeste"), 2, TRUE)')
ORDER BY s.SCORE DESC;
GO

```

```

SELECT      s.matched_document_key, s.SCORE, d.Questão
FROM SEMANTICSIMILARITYTABLE(BD_Resposta, Resposta, 1) AS s
        INNER JOIN BD_Resposta AS d
        ON s.matched_document_key = d.DocId
WHERE SCORE > 0.6 and
CONTAINS(Resposta,'FORMSOF(INFLECTIONAL,contribuir) ')
ORDER BY s.SCORE DESC;
GO

```

-- Localizar o TSAURUSFILE (o caminho pode variar um pouco de acordo com a versão do SQL Server utilizada).

C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS17\MSSQL\FTData\tsptb

-- Arquivo tsptb – Português (Brasil)

<XML ID="Microsoft Search Thesaurus">

<!-- Commented out

```

    <thesaurus xmlns="x-schema:tsSchema.xml">
<diacritics_sensitive>0</diacritics_sensitive>
    <expansion>
        <sub>Internet Explorer</sub>
        <sub>IE</sub>
        <sub>IE5</sub>
    </expansion>
    <replacement>

```

```

    <pat>NT5</pat>
    <pat>W2K</pat>
    <sub>Windows 2012</sub>
</replacement>
<expansion>
    <sub>run</sub>
    <sub>jog</sub>
</expansion>
</thesaurus>
-->
</XML>

```

```

-- Carregar um arquivo de dicionário de sinônimos
EXEC sys.sp_fulltext_load_thesaurus_file 1046;

```

```

SELECT      s.matched_document_key, s.SCORE, d.Questão
FROM SEMANTICSIMILARITYTABLE(BD_Resposta, Resposta, 1) AS s
      INNER JOIN BD_Resposta AS d
      ON s.matched_document_key = d.DocId
WHERE SCORE > 0.6 and
      CONTAINS(Resposta,'FORMSOF(THESAURUS,auxilie)')
ORDER BY s.SCORE DESC;
GO

```

```

SELECT s.keyphrase, s.SCORE
FROM SEMANTICSIMILARITYDETAILSTABLE(BD_Resposta, Resposta, 4,
Resposta, 5) AS s
WHERE S.SCORE > 0.6
ORDER BY s.SCORE DESC;
GO

```

```
SELECT COUNT(s.keyphrase) 'Qtd de palavras em comum'
FROM SEMANTICSIMILARITYDETAILSTABLE(BD_Resposta, Resposta, 1,
Resposta, 5) AS s
WHERE s.SCORE > 0.3
ORDER BY COUNT(s.keyphrase);
GO
```