Consultas analíticas por similaridade em SGBD Relacionais

Antônio Lívio C. de Mendonça[1], Maria Camila N. Barioni[1], Humberto Razente[1]

[1]Universidade Federal de Uberlândia
Av. João Naves de Ávila, 2121, Uberlândia, MG

{antonio.mendonca,camila.barioni,humberto.razente}@ufu.br

Abstract. *The wide variety of complex data produced in recent times, for which equality searches have little relevance, has led to the development of similarity-based query operations. However, few works deal with similarity in the context of online analytical processing. This work presents an approach for executing ad hoc analytical queries, where the grouping criteria can be based on similarity, especially in the metric, spatial, and time contexts. The prototype developed to validate and illustrate the concepts shows that this criterion can be executed at different levels of granularity, through features of the standard SQL language in Relational Database Management Systems.*

Resumo. *A grande variedade de dados complexos produzidos nos últimos tempos, para os quais as buscas por igualdade tem pouca utilidade, levou ao desenvolvimento de operações de consulta baseadas em similaridade. Entretanto, poucos trabalhos consideram o tratamento da similaridade no contexto de processamento analítico online. Neste trabalho[1] apresenta-se uma abordagem para a execução de consultas ad hoc analíticas, onde o critério de agrupamento pode ser baseado na similaridade, em especial nos contextos métrico, espacial e temporal. O protótipo desenvolvido para validar e ilustrar os conceitos mostra que esse critério pode ser executado em diferentes níveis de granularidade, por meio de recursos da linguagem padrão SQL em Sistemas de Gerenciamento de Bancos de Dados Relacionais.*

1. Introdução

Atualmente as pessoas interagem com diversos sistemas que coletam dados complexos para os quais é mais interessante realizar buscas por similaridade [Matiazzo et al. 2023] envolvendo dados vetoriais, multimídia, sequências genômicas, séries temporais, coordenadas geográficas, rotas, dados de sensores, entre outros. Nesse contexto, são desejadas técnicas e estratégias eficientes para armazenamento, organização, recuperação e análise desses dados. O modelo relacional e a linguagem SQL são amplamente utilizados na manipulação desses dados [Stonebraker and Pavlo 2024].

Tarefas de análise de dados usualmente utilizam sistemas OLAP (*Online Analytical Processing*) pois estes provêem um modo eficiente de organização de grandes volumes de dados para produção de relatórios que apoiam a tomada de decisões. Entre as suas vantagens estão a rapidez e facilidade em lidar com grandes volumes de dados. A principal

[1]Realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

característica do OLAP é a multidimensionalidade. O cubo (ou hipercubo) de dados é usado para visualização das várias dimensões de interesse que podem ser exploradas de forma hierárquica para tomada de decisões [Abelló and Romero 2018].

Inicialmente a agregação de dados para criação de cubos multidimensionais baseava-se apenas no critério de igualdade [ISO 1992] e não permitia a criação de histogramas[2] ou a computação de sub-totais das operações *drill-down* e *roll-up* [Gray et al. 1997]. Estas características estão atualmente incorporadas na linguagem padrão [ISO 2023] e são implementadas por vários fabricantes. Nesses sistemas os domínios de valores em geral são cadeias de caracteres, números ou datas.

Entre os dados comumente manipulados em Sistemas de Gerenciamento de Bancos de Dados (SGBD) Relacionais, para os quais os agrupamentos por similaridade podem ser desejáveis, destacam-se textos, datas, dados espaciais, e conjuntos de atributos para os quais pode-se definir uma métrica. Os trabalhos correlatos sugerem alterações na linguagem padrão SQL para realização dessas consultas. Neste trabalho apresenta-se uma abordagem para a execução de consultas *ad hoc* analíticas, onde o critério de agrupamento pode ser baseado na similaridade por meio de funções definidas pelo usuário (UDF) na linguagem padrão SQL. O uso de funções na especificação de uma cláusula GROUP BY passou a permitir o tratamento da granularidade em consultas OLAP, além de permitir novas aplicações, como a criação de agrupamentos por similaridade.

2. Conceitos Fundamentais

A similaridade pode ser empregada sobre dados em diferentes contextos. Dados temporais podem ser comparados diretamente pela diferença de seus valores, expressos numericamente como unidades baseadas em 1 dia. Textos curtos podem ser comparados pela distância de edição, que corresponde ao número de inserções, remoções ou trocas para transformar uma cadeia de caracteres em outra, por meio de ontologias, entre outras formas. A similaridade em dados espaciais pode ser computada por funções de distância ou pela similaridade das geometrias, por exemplo, por operações topológicas (contém, está contido, interseção, etc).

Já a noção de similaridade em dados multimídia é geralmente obtida a partir da transformação desses dados em vetores de características, por meio de algoritmos específicos de cada domínio de aplicação, sendo que a similaridade (ou dissimilaridade) entre um par desses vetores é computada por uma função de distância em um espaço métrico. O domínio dos dados $\mathbb{S}$ e a função de distância $\delta()$ definem um espaço métrico, onde $\delta()$ satisfaz as seguintes propriedades para quaisquer elementos x, y, $z \in \mathbb{S}$: $\delta(x,x) = 0$ (*identidade*); $\delta(x,y) = \delta(y,x)$ (*simetria*); $0 \leq \delta(x,y) < \infty$ (*não-negatividade*); $\delta(x,y) \leq \delta(x,z) + \delta(z,y)$ (*desigualdade triangular*).

Entre as funções de distância amplamente utilizadas estão as da família Minkowski ou métricas L_p, que são aplicadas em domínios multidimensionais, dentre elas as distâncias Manhattan (L_1), Euclidiana (L_2) e Chebychev (L_∞). A desigualdade triangular é uma propriedade importante pois permite a criação de algoritmos e estruturas de dados que organizam ou particionam o espaço dos dados. Tais algoritmos permitem,

[2]Um histograma é resultado de uma função aplicada ao atributo de agregação com o objetivo de lidar com uma granularidade arbitrária, por exemplo, transformando uma instância de tempo em semana, quinzena, trimestre, estação do ano, etc.

por exemplo, na realização da busca de elementos similares a um elemento de referência, descartar partições em conjuntos de dados previamente particionados, otimizando assim a execução da tarefa. Extensas revisões sobre a indexação métrica podem ser encontradas em [Samet 2006, Chen et al. 2023].

O agrupamento de elementos similares é um problema com grande utilidade em diversos tipos de análise ou aplicações. O objetivo de um processo de agrupamento é dividir os elementos de um conjunto de dados em grupos de elementos similares, de modo que cada grupo seja composto de elementos mais similares entre si e dissimilares aos elementos de outros grupos [Jain 2010]. Os métodos de agrupamento podem ser divididos em hierárquicos e por particionamento. Os algoritmos hierárquicos produzem uma hierarquia que consiste de vários níveis de partições de elementos aninhadas. Os algoritmos que fazem particionamento tentam encontrar o melhor conjunto de k partições de dados, criando apenas um nível de particionamento que divide todos os elementos em grupos. Diversos algoritmos foram propostos nas últimas décadas em diversos contextos, como identificação de agrupamentos em fluxos de dados contínuos, índices dinâmicos, e dados estáticos [Ezugwu et al. 2022].

A integração de operações de consultas por similaridade em dados complexos em SGBD Relacionais tem sido abordada pelos trabalhos correlatos da área considerando várias frentes de pesquisa, entre elas: a implementação de operações para execução de consultas por similaridade [Gray et al. 1997, Barioni et al. 2009, Kaster et al. 2010, Tang et al. 2016, Lu et al. 2017, Kim et al. 2020]; e a otimização de operações elementares [Barioni et al. 2008, Razente et al. 2008, Oliveira et al. 2023, Eleutério et al. 2024]. O foco do trabalho descrito neste artigo está relacionado com a primeira vertente de pesquisa.

A integração de consultas por similaridade em SGBD relacionais é uma tarefa difícil uma vez que o modelo relacional não prevê a manipulação de vetores e matrizes como tipos nativos de dados [Garcia-Alvarado and Ordonez 2015]. Com isso, os trabalhos relacionados da área têm abordado o uso de UDF [Kaster et al. 2010, Kim et al. 2020] ou com propostas de extensão da lingugagem SQL [Silva et al. 2009b, Tang et al. 2016]. A maioria dos trabalhos que utiliza UDF se concentra na disponibilização de operações por similaridade básicas como consultas por abrangência e consultas aos k-vizinhos mais próximos. Poucos trabalhos têm abordado a disponibilização de consultas análiticas sobre dados que requerem operações de comparação por similaridade.

A agregação por similaridade [Silva et al. 2009a] para consultas analíticas vem sendo estudada em espaços métricos e em espaços multidimensionais [Tang et al. 2016], e otimizadas com processamento distribuído [Silva et al. 2019]. O trabalho de [Iqbal et al. 2022] apresenta um formalismo para a agregação por similaridade de textos, dados espaciais e temporais para permitir a visualização de cubos multidimensionais de dados, para processamento analítico online (OLAP). Para o tratamento de texto, utiliza a técnica *bag-of-words* e propõe o uso de ontologias para lidar com a granularidade dos agrupamentos. A informação espacial é uma localização com latitude e longitude. O tempo é expresso como um instante preciso em alguma resolução (por exemplo, segundos), que pode ser agrupado com base na semântica (dia, mês, ano, semana, quinzena, trimestre, semestre, estação do ano, entre outros). As dimensões texto-espaço-tempo podem ser utilizadas para criação de hierarquias empregadas nas operações OLAP de aumento

ou diminuição da granularidade da similaridade (*roll-up* ou *drill-down*).

Os trabalhos correlatos abrem caminho para uma abordagem conjunta do que é apresentado em [Silva et al. 2009a] e [Iqbal et al. 2022] usando a linguagem SQL padrão. Enquanto [Silva et al. 2009a] propõe uma abordagem de extensão da linguagem SQL, este trabalho propõe o uso de UDF para definição de uma função de agregração por similaridade para particionar os dados em conjuntos de dados similares entre si. Em [Iqbal et al. 2022] o processamento dos agrupamentos é feito em aplicações separadas do bando de dados, e neste apresenta-se uma solução implementada no SGBD.

3. Group-By Métrico, Espacial e Temporal

O agrupamento por similaridade sobre uma relação R gera grupos de tuplas não sobrepostos com base em uma função de agregação $agr\_sim\_metr()$, que recebe como parâmetros os atributos sobre os quais irá operar e um limiar de distância ($dist$), e retorna um identificador de segmento para cada grupo, de modo que tuplas que compartilham um mesmo identificador pertencem ao mesmo grupo.

A Listagem 1 apresenta um exemplo de utilização de uma função de similaridade, $sim\_metr()$, em uma consulta com agrupamento e a Figura 1(a) apresenta uma intuição da semântica do agrupamento por similaridade.

Listagem 1. Agrupamento por similaridade em SQL.

```
SELECT metrico, count(*)
  FROM relacaoR
GROUP BY sim_metr(atrs, dist) as metrico;
```

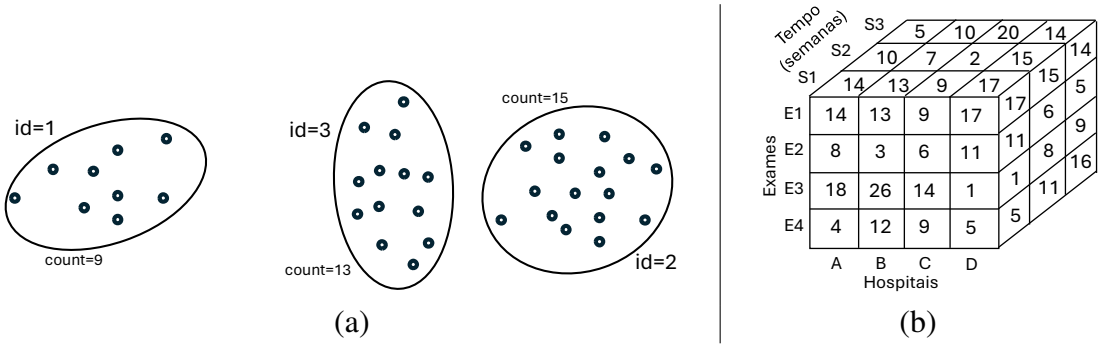

Figura 1. Exemplo de uso do operador SGB.

As consultas que contêm cláusulas GROUP BY com dois ou mais atributos (dimensões) permitem a criação de tabelas cruzadas (*cross-tabs*) ou cubos de dados (*data cubes*) dos fatos, para os quais em geral aplicam-se funções de agregação (sum, min, max, agv, count) sobre seus atributos. A Figura 1(b) apresenta a visualização tridimensional do resultado da Listagem 2, que permite agregar a contagem de tuplas pela similaridade dos exames, pelos hospitais espacialmente, e no tempo por semana. A granularidade é definida pelo limiar de distância para a formação dos grupos de tuplas, pois expande ou restringe o espaço de similaridade.

Listagem 2. Agrupamento por similaridade em SQL em 3 dimensões.

```
SELECT Exames, Hospitais, Tempo, count(*)
  FROM Fatos
GROUP BY sim_metr(hem, vcm, hcm, chcm, 5) as Exames,
         sim_esp(lat, long, 3) as Hospitais,
         semana(data) as Tempo;
```

Após a execução da consulta, operações *slice* e *dice* podem ser realizadas pela ferramenta de visualização empregada para exibição do cubo. As operações *roll-up* e *drill-down* são obtidas com a re-execução das consultas com novos limiares de distância.

Para a validação da abordagem proposta foi considerado o conjunto de dados $breastcancer$ [Kelly et al. 2024] por conter vários atributos numéricos que podem ser tratados como um vetor de características que podem ser comparados pela sua dissimilaridade. Além disso, foram adicionados dois atributos: um espacial e um temporal, que representam respectivamente o local e a data de coleta (ou registro) do exame. Apesar de não ser uma base de dados orgânica, o mais importante neste trabalho é o formato dos dados e as dimensões Métrica (dados complexos), Espacial (lat/lon ou área) e Temporal (data). Em um cenário real onde os dados são provenientes de um sistema hospitalar, os mesmos já teriam esses atributos, como no conjunto de tweets usado em [Iqbal et al. 2022]. Apesar de ter sido realizada no PostgreSQL, a implementação em SGBD relacionais de vários fornecedores que implementam os recursos descritos da linguagem SQL padrão pode ser obtida pela tradução das funções, resguardadas as diferenças nas sintaxes das suas linguagens procedurais.

4. Funções para Agrupamentos por Similaridade

O Algoritmo 1 descreve a função $sim\_metr()$. Essa função percorre as tuplas da relação listada na cláusula FROM ($breastcancer$) verificando se o ponto já foi atribuído a algum centroid (grupo). Para isso, a função realiza consultas nas tabelas temporárias $centroids$ e $centroid\_match$ que foram criadas para guardar, respectivamente, os grupos definidos pelo algoritmo de agrupamento e a relação entre as tuplas da tabela $breastcancer$ e $centroids$. O uso de tabelas temporárias para armazenar os resultados intermediários do agrupamento é importante para manter o isolamento de outras transações.

Na linha 11 do Algoritmo 1 a função $agr\_sim\_metr()$ é invocada para particionar os dados. Caso a tupla em questão ainda não tenha sido incluída em algum agrupamento, o Algoritmo 2 realiza a inclusão. Uma vez que a tupla é adicionada a um grupo, o seu centroid é atualizado para refletir a mudança no grupo (linha 5). A equação foi omitida por questão de simplicidade mas a implementação computa a média entre o centroid atual e o ponto ingressante.

Vale ressaltar que $agr\_sim\_metr()$ descrita no Algoritmo 2 é uma simplificação do algoritmo *k-médias* adaptada para a estratégia adotada em [Silva et al. 2009a]. Por se tratar de uma UDF, pode ser refinada e implementada conforme necessidade do usuário ou do domínio dos dados.

As duas funções utilizaram a biblioteca $Postgis$ para lidar com pontos em 3 dimensões mas não se trata de uma depêndencia. Os cálculos de distância podem, e foram

Algoritmo 1: sim_metr()

Input: $id,\ x,\ y,\ z,\ dist$
Output: c_id /* id do centroid */

Declare c_id (int), $ponto$ (geometry)
$ponto \leftarrow$ makePoint$(x,\ y,\ z)$
if *(count(*) FROM* $centroids$*)* $= 0$ **then**
  $centroids \leftarrow (1, ponto)$
  $c\_id \leftarrow 1$
else
  $c\_id \leftarrow$ SELECT $cent\_id$ FROM $centroid\_match$ WHERE $bc\_id = id$
  if $c\_id \neq$ *NULL* **then**
    return c_id
  else
    $c\_id \leftarrow$ agr_sim_metr$(ponto,\ dist)$
$centroid\_match \leftarrow (id, c\_id)$
return c_id

Algoritmo 2: agr_sim_metr()

Input: $ponto,\ dist$
Output: c_id /* id do centroid */

Declare i (int), l ($centroids$ linha)
$i \leftarrow$ SELECT $count(*)$ FROM $centroids$
for $l \leftarrow$ *SELECT * FROM* $centroids$ **do**
  if *ST_3DDWinthin*$(g.centro,\ ponto,\ dist)$ **then**
    UPDATE $centroids$ SET c_loc = ST_MakePoint() WHERE $id = l.id$
$centroids \leftarrow (i + 1,\ ponto)$
return c_id

(em alguns casos), implementados sem usar esse recurso. Os algoritmos foram implementados[3] no PostgreSQL versão 16.3.

Um exemplo de consulta nesse contexto foi apresentado na Listagem 2. Ela poderia responder a gestores em saúde como se agrupam pacientes com uma determinada enfermidade adequando ações às necessidades de cada região.

5. Conclusões

Neste trabalho foi apresentada uma estratégia para realização de agrupamentos por similaridade para consultas analíticas utilizando a linguagem SQL padrão. O agrupamento por similaridade permite a criação de tabelas cruzadas ou cubos de dados dos fatos, permitindo a sua visualização multidimensional e as operações OLAP tradicionais. Foi apresentada uma solução na linguagem PL/pgSQL, por meio de funções definidas pelo usuário, que foi validada considerando um conjunto de dados contendo dimensões métrica, temporal e espacial.

[3] https://github.com/liviomendonca/sqlsim

Referências

Abelló, A. and Romero, O. (2018). Online analytical processing. In *Encyclopedia of Database Systems*, pages 2558–2563. Springer. doi:10.1007/978-1-4614-8265-9_252.

Barioni, M. C. N., Razente, H., Traina, A., and Traina-Jr, C. (2008). Accelerating k-medoid-based algorithms through metric access methods. *J. Syst. Softw.*, 81(3):343–355. doi:10.1016/J.JSS.2007.06.019.

Barioni, M. C. N., Razente, H., Traina, A., and Traina-Jr., C. (2009). Seamlessly integrating similarity queries in SQL. *Softw. Pract. Exp.*, 39(4):355–384. doi:10.1002/SPE.898.

Chen, L., Gao, Y., Song, X., Li, Z., Zhu, Y., Miao, X., and Jensen, C. S. (2023). Indexing metric spaces for exact similarity search. *ACM Comput. Surv.*, 55(6):128:1–128:39. doi:10.1145/3534963.

Eleutério, I., Cazzolato, M., Gutierrez, M. A., Teixeira, L., Traina, A., and Traina-Jr, C. (2024). Migue-sim: Speeding up similarity queries with native rdbms resources. In *Symp. Applied Computing (SAC)*, pages 321–328. doi:10.1145/3605098.3636019.

Ezugwu, A., Ikotun, A., Oyelade, O., Abualigah, L., Agushaka, J., Eke, C., and Akinyelu, A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Eng. Appl. Artif. Intell.*, 110:104743. doi:10.1016/J.ENGAPPAI.2022.104743.

Garcia-Alvarado, C. and Ordonez, C. (2015). Clustering binary cube dimensions to compute relaxed GROUP BY aggregations. *Inf. Syst.*, 53:41–59. doi:10.1016/j.is.2014.12.008.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. (1997). Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub totals. *Data Min. Knowl. Discov.*, 1(1):29–53. doi:10.1023/A:1009726021843.

Iqbal, M., Lissandrini, M., and Pedersen, T. B. (2022). A foundation for spatio-textual-temporal cube analytics. *Inf. Syst.*, 108:102009. doi:10.1016/j.is.2022.102009.

ISO (1992). *ISO/IEC 9075:1992: Information technology — Database languages — SQL*. International Org. Standardization. https://www.iso.org/standard/16663.html.

ISO (2023). *ISO/IEC 9075:2023: Information technology — Database languages — SQL*. International Org. Standardization. https://www.iso.org/standard/76583.html.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett.*, 31(8):651–666. doi:10.1016/J.PATREC.2009.09.011.

Kaster, D. S., Bugatti, P. H., Traina, A. J. M., and Traina-Jr, C. (2010). FMI-SiR: A flexible and efficient module for similarity searching on Oracle database. *J. Inf. Data Manag.*, 1(2):229–244. doi:10.5753/jidm.2010.1263 .

Kelly, M., Longjohn, R., and Nottingham, K. (2024). The UCI Machine Learning Repository. https://archive.ics.uci.edu.

Kim, T., Li, W., Behm, A., Cetindil, I., Vernica, R., Borkar, V. R., Carey, M. J., and Li, C. (2020). Similarity query support in big data management systems. *Inf. Syst.*, 88. doi:10.1016/J.IS.2019.101455.

Lu, W., Hou, J., Yan, Y., Zhang, M., Du, X., and Moscibroda, T. (2017). MSQL: efficient similarity search in metric spaces using SQL. *VLDB J.*, 26(6):829–854. doi:10.1007/s00778-017-0481-6.

Matiazzo, M. A. L., de Castro-Silva, V., Oyamada, R. S., and Kaster, D. S. (2023). The dataset-similarity-based approach to select datasets for evaluation in similarity retrieval. In *Intl Conf. Similarity Search and Applications (SISAP)*, volume 14289 of *LNCS*, pages 125–132. Springer. doi:10.1007/978-3-031-46994-7_11.

Oliveira, W. D., Lauton, A. J. C., Traina-Jr, C., and Santos, L. F. D. (2023). Similarity grouping by influence: Exploring result diversification in similarity group-by operators. In *Simpósio Brasileiro de Bancos de Dados (SBBD)*, pages 402–407. SBC. doi:10.5753/sbbd.2023.233430.

Razente, H., Barioni, M. C. N., Traina, A., Faloutsos, C., and Traina-Jr, C. (2008). A novel optimization approach to efficiently process aggregate similarity queries in metric access methods. In *Int'l Conf. Information and Knowledge Management (CIKM)*, pages 193–202. ACM. doi:10.1145/1458082.1458110.

Samet, H. (2006). *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann. San Francisco, CA.

Silva, Y. N., Aref, W. G., and Ali, M. H. (2009a). Similarity group-by. In *Int'l Conf. Data Engineering (ICDE)*, pages 904–915. IEEE. doi:10.1109/ICDE.2009.113.

Silva, Y. N., Arshad, M. U., and Aref, W. G. (2009b). Exploiting similarity-aware grouping in decision support systems. In *Int'l Conf. Extending Database Technology (EDBT)*, volume 360, pages 1144–1147. ACM. doi:10.1145/1516360.1516499.

Silva, Y. N., Sandoval, M., Prado, D., Wallace, X., and Rong, C. (2019). Similarity grouping in big data systems. In *Intl Conf. Similarity Search and Applications (SISAP)*, volume 11807 of *LNCS*, pages 212–220. Springer. doi:10.1007/978-3-030-32047-8_19.

Stonebraker, M. and Pavlo, A. (2024). What goes around comes around... and around... *SIGMOD Rec.*, 53(2):21–37.

Tang, M., Tahboub, R. Y., Aref, W. G., Atallah, M. J., Malluhi, Q. M., Ouzzani, M., and Silva, Y. N. (2016). Similarity group-by operators for multi-dimensional relational data. *IEEE Trans. Knowl. Data Eng.*, 28(2):510–523. doi:10.1109/TKDE.2015.2480400.