

Ferramenta para Extração de Grafo a partir de Modelo Relacional de Banco de Dados

Pedro F. J. Duarte¹, Daves M. S. Martins²

¹Instituto Federal do Sudeste de Minas Gerais (IF Sudeste MG)
36.080-001 – Juiz de Fora – MG – Brasil

36.080-

²Instituto Federal do Sudeste de Minas Gerais (IF Sudeste MG)
36.080-001 – Juiz de Fora – MG – Brasil

pedrofjduarte@gmail.com, daves.martins@ifsudestemg.edu.br

Abstract. *This article brings a comparison between two database models that are relational and graph-oriented, bringing discussions and ways to store data in both models. Proposing situations in which a database in the relational model is viable its migration to the graph-oriented one and when it is not feasible. Bringing situations where the joins caused by the relational model due to relationships do not pay off to solve certain types of problems. The works selected in the review, how the selection process of the works was done and a comparison between the selected works. With this we will also propose the conversion of the relational model to the graph. The process of developing the conversion tool for the graph model. Results obtained with the conversion and considerations about the results obtained after the experiment.*

Resumo. *Este artigo traz uma comparação entre dois modelos de banco de dados que é o relacional e o orientado a grafo, trazendo discussões e formas de armazenar os dados em ambos os modelos. Propondo situações em que um banco no modelo relacional é viável a sua migração para o orientado a grafo e quando não é viável. Trazendo situações em que as junções causadas pelo modelo relacional devido aos relacionamentos não compensam para resolver determinados tipos de problemas. Os trabalhos selecionados na revisão, como foi feito o processo de seleção dos trabalhos e um comparativo entre os trabalhos selecionados. Com isso também vamos propor a conversão do modelo relacional para o grafo. O processo de desenvolvimento da ferramenta de conversão para o modelo grafo. Resultados obtidos com a conversão e as considerações sobre os resultados obtidos após o experimento.*

1. Introdução

O modelo de banco de dados relacional é uma excelente solução, maduro e amplamente adotado pelo mercado. Mas pode haver casos em que o modelo relacional não seja uma boa solução para resolver determinados problemas. Devido a baixa eficiência causada pelos relacionamentos ao trazer informações em que é necessário a junção entre tabelas.

Para lidar com volumes de dados gerados e processados por redes sociais, computação na nuvem e Big Data hoje em dia, é necessário que o modelo do banco de dados não se torne um gargalo ao lidar com esses dados.

Nesse momento entra o modelo Grafo como uma solução para determinados problemas em que o relacional encontra dificuldades e problemas para lidar.

Segundo (FOWLER e SADALAGE, 2012), os casos de uso do modelo orientado a grafo consiste-se nas seguintes: aplicações do domínio social com as redes sociais, aplicações de domínio espacial, como os problemas de roteamento e transporte; e comercio, como os motores de recomendação de uma plataforma de e-commerce.

Vamos conceituar a parte de Grafo para que fique claro no momento de aplicá-lo em banco de dados.

Fazer comparativos entre os dois modelos, abordando vantagens e desvantagens de ambos os modelos.

Vai ser proposto uma conversão do modelo relacional para o modelo grafo, para assim podermos analisar se realmente foi vantajosa essa migração e se foi viável.

2. Tipos de Banco de Dados

O banco de dados é de grande importância para qualquer organização, seja lá qual for o setor e a atividade que demande o armazenamento de dados. Além de ser fundamental para auxiliar os processos e atividades de trabalho do dia a dia, os bancos de dados proporcionam as organizações uma forma de obterem vantagens competitivas no mercado e na tomada de decisões do ponto de vista estratégico. Obtendo informações e gerando conhecimento através da análise desses dados armazenados. Segundo MACHADO (1996, p.14) os dados são baseados no princípio de serem estáveis no decorrer da vida de uma determinada empresa ou organização, possuindo características voláteis por refletir uma realidade a ser automatizada e não apenas um momento a ser modificado.

Diante desse cenário temos tipos de banco de dados que podem ser adotados. O banco de dados Relacional é um modelo onde os dados são armazenados em tabelas, podendo essas tabelas terem ou não ligações uma com a outra. Já o modelo não relacional orientado a Grafo o banco é fundamentado na estrutura de um grafo. Que é representado através de pontos e os caminhos que fazem a comunicação entre esses pontos. Na figura 1 é possível observar os 2 modelos representados.

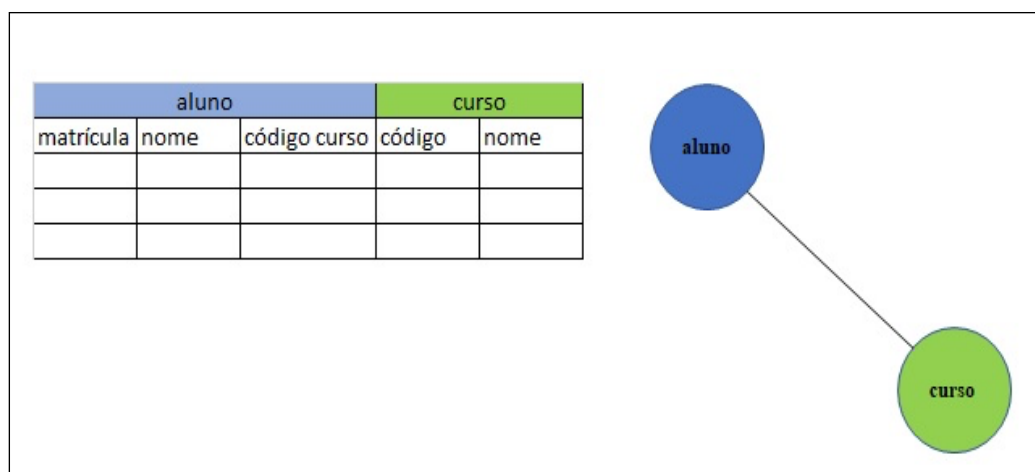


Figura 1. Modelo em tabela e Grafo

Ao lado esquerdo da figura temos a representação do modelo relacional, com as tabelas aluno e curso. Do lado direito temos o modelo grafo, onde aluno e curso são representados por círculos e ligados um ao outro através de uma reta.

Devido as características e particularidades de cada modelo de banco de dados as empresas tendem a optar por modelos de banco que se adéquem ao seu negócio ou problema levando em conta facilidade que o modelo tem para lidar com o problema, desempenho e espaço para armazenamento.

Com o relacional sendo um modelo popular e amplamente adotado como solução ao longo do tempo, algumas empresas podem sentir a necessidade de mudarem algumas das suas soluções que adotaram o modelo relacional para o modelo grafo, justamente para se adequar a um modelo mais indicado para lidar com o problema.

2.1. Relacional

O modelo relacional que já é bastante conhecido pela sua forma de armazenar os dados em tabelas e estabelecer relacionamentos entre as tabelas é amplamente adotado pelo mercado devido sua maturidade e a forma de lidar com os dados.

Mas um fator que impacta bastante nesse modelo e está relacionado ao desempenho é o relacionamento entre tabelas. O custo para trazer informações em que os dados estão em tabelas diferentes se torna elevado ao realizar essas junções.

Segundo EICH e MISHRA (1992, p.21), a junção é uma das operações fundamentais para consultas em BDs relacionais. Ela permite associar tuplas de tabelas diferentes, permitindo a recuperação de dados correlacionados presentes em mais de uma tabela. A implementação dessa operação foi baseada no produto cartesiano entre tabelas, e é uma operação custosa para os SGBDs.

Um exemplo pode ser visto na figura 2, onde temos três tabelas que representam a solução para um negócio de vendas de produtos.



Figura 2. Modelo relacional de banco de dados de vendas

Na tabela de vendas armazena-se o código da venda e a data que ocorreu aquela venda, na tabela de produtos guardamos o código de barras do produto, nome, preço e quantidade. Já na tabela itens_vendas, que possui relacionamentos com as tabelas vendas e produtos, encontra-se o código dos itens de venda, a quantidade e o preço. Pelo fato da tabela itens_vendas se relacionar com outras duas tabelas fazem com que ela guarde referências para as mesmas, com isso na tabela itens_vendas também vão ser armazenados o código de barras do produto e o código da venda.

Usando esse mesmo modelo apresentado na figura 2 é possível levantar algumas situações. Se for preciso buscar informações de todos os itens vendidos e exibir o nome de cada produto, seria necessário fazer a junção das tabelas produtos e itens_vendas para trazer essas informações.

Outro caso é que seria interessante buscar todos os itens vendidos em um determinado mês incluindo o nome dos produtos, nessa situação já seria necessário fazer duas junções envolvendo as três tabelas.

No modelo relacional, essas junções que ocorrem entre tabelas para trazer determinadas informações, como foi visto anteriormente, é que fazem com que os relacionamentos muitas vezes tenham uma baixa eficiência.

2.2. Grafos

No modelo orientado a Grafo a estrutura e organização dos dados é fundamentado nos conceitos de Grafo. Então para se ter uma melhor compreensão desse modelo vamos conceituar primeiramente o que é um Grafo e sua estrutura.

Um Grafo é uma estrutura abstrata que representa um conjunto de elementos denominados vértices e suas relações de interdependência ou arestas (GOLDBARG, 2012, p.2).

Grafo é basicamente formado por pontos que são conhecidos como vértices e esses vértices podem ou não ter uma reta que liga um vértice ao outro, que é conhecida como aresta.

Para exemplificar melhor os conceitos abordados até o momento pode-se observar na figura 3 onde a representação de um grafo para demonstrar o caminho entre dois municípios pode ser visto.

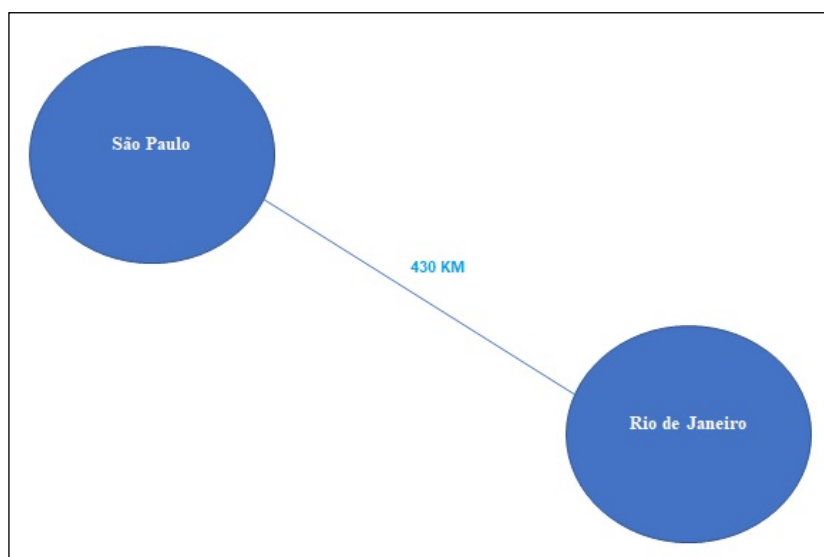


Figura 3. Grafo da ligação entre dois municípios

No Grafo temos dois vértices que representam os municípios de São Paulo e Rio de Janeiro e um caminho que liga os dois municípios que é uma aresta. Observa-se que no

Grafo através da aresta temos um dado importante, que é 430 KM, que quer dizer que nesse caminho entre os dois municípios temos 430 quilômetros. Com isso através de uma representação de um Grafo consegue-se obter essas informações.

Uma outra maneira de representar o Grafo é em relação à sua orientação. No exemplo demonstrado na figura anterior, um Grafo é representado de forma não orientada, ou seja. Ele não tem um único sentido definido de qual ponto é a partida até o ponto de chegada.

Na forma orientada representa-se o Grafo com uma aresta em forma de seta, indicando o sentido do ponto de partida até o ponto final.

Segundo (GOLDBARG, 2012, p. 8) um grafo é dito direcionado ou orientado quando o sentido das ligações entre os vértices é importante.

Veja na figura abaixo como o grafo que liga os municípios de São Paulo e Rio de Janeiro foi representado agora de forma direcionada.

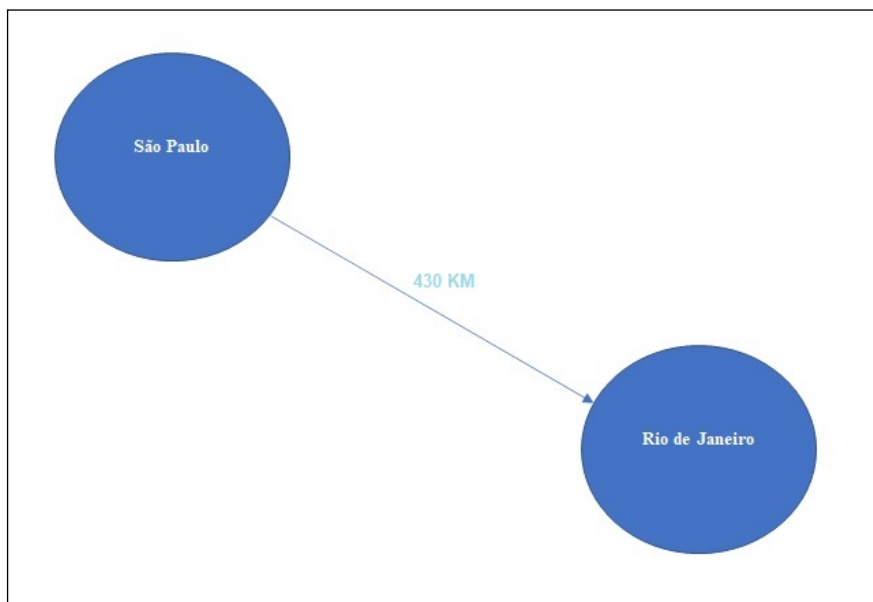


Figura 4. Grafo da ligação entre dois municípios de forma orientada

Note que agora passamos a dar um maior sentido a informação, que além de dizer que entre São Paulo e Rio de Janeiro temos 430 quilômetros, deixamos claro que estamos saindo de São Paulo em direção ao Rio de Janeiro.

3. Revisão sistemática

A revisão sistemática foi feita com o objetivo de fazer o levantamento de trabalhos relacionados à conversão de modelos de banco de dados relacionais para modelos orientados a grafo.

Foram feitas buscas de trabalhos via web no Google Scholar, utilizando as palavras chaves: converter, extração de esquema, banco de dados de grafos e modelo entidade-relacionamento. Nas buscas o idioma considerado foram os trabalhos em português.

Após as buscas foram retornados 8 trabalhos e utilizando o critério de ser um trabalho em que é feita a conversão ou extração de modelo, do relacional de banco de dados para o orientado a Grafo. Foram feitas leituras nos resumos dos trabalhos e selecionando apenas os que estavam de acordo com o critério definido. Alguns trabalhos também não foram selecionados por estarem redundantes aos já selecionados. Trabalhos de conversão de modelos não relacionais e que não tinha haver com conversão de relacional para o modelo grafo também não forma considerados. Então, dos 8 trabalhos retornados, foram selecionados 4.

Trabalhos selecionados	Características
Base de Dados Orientada a Grafos: Um Experimento aplicado a Análise Social.	Nesse trabalho foi feita uma modelagem e um implementação de um banco de dados de grafo para identificar diferenças em relação ao modelo relacional.
Análise de Projetos de Banco de Dados: Modelo Relacional vs. Modelo em Grafos.	Este apresentou diferenças no processo de modelagem de banco de dados relacionais e orientado a grafos. E um processo mais estruturado para o modelo grafo.
Modelagem em Grafos a partir de Bancos de Dados Relacionais.	Propôs heurísticas para o mapeamento de modelos relacionais para os modelos grafo.
Projeto Lógico de Banco de Dados NoSql de Grafos a partir de um modelo conceitual baseado no modelo Entidade-Relacionamento.	Proposta de um projeto de banco de dados NoSql a partir de modelo relacional.

Tabela 1. Quadro comparativo dos trabalhos selecionados

Durante o trabalho foi feito o processo de estudos para escolher a ferramenta de modelagem do banco relacional e que fosse possível fazer a conversão do modelo a partir do arquivo salvo e o formato do mesmo. Forma feitos experimentos para identificar e entender com qual ferramenta seria possível fazer a conversão.

Após escolha da ferramenta de modelagem relacional com o uso da linguagem de programação Java foi feita a leitura de arquivos de formato xml, com os modelos relacionais e a implementação da conversão do relacional para grafo, lógica e persistência do mesmo em arquivo de armazenamento de grafo.

A ideia central da ferramenta foi converter as tabelas em vértices e os relacionamentos em aresta e transformar o modelo relacional em um modelo grafo.

Foram realizados testes com a ferramenta de conversão desenvolvida, com diversos modelos.

Uma serie de experimentos com ferramentas de leitura de arquivo de formato de grafo foram feitas, até conseguir exibir o grafo visualmente, que foi possível em uma ferramenta online de construção e leitura de grafo.

Durante a realização do trabalho também foram feitos o uso de livros para auxiliar na realização do mesmo e consulta de materiais e documentações online para resolver dificuldades técnica encontradas durante a implementação da ferramenta.

Escolha de um modelo relacional para fazer o experimento de conversão relacional para grafo através da ferramenta desenvolvida.

4. Comparativo Relacional x Grafos

Fazendo o comparativo entre os modelos relacional e grafo vai ser possível compreender e deixar claro que não são todos os bancos que estão no modelo relacional precisam sofrer uma migração para o modelo orientado a Grafos.

Considere que temos um sistema de cadastramento de rotas para realizar entregas de uma determinada transportadora e o banco de dados está modelado em uma estrutura relacional, conforme e mostrado na figura 5.

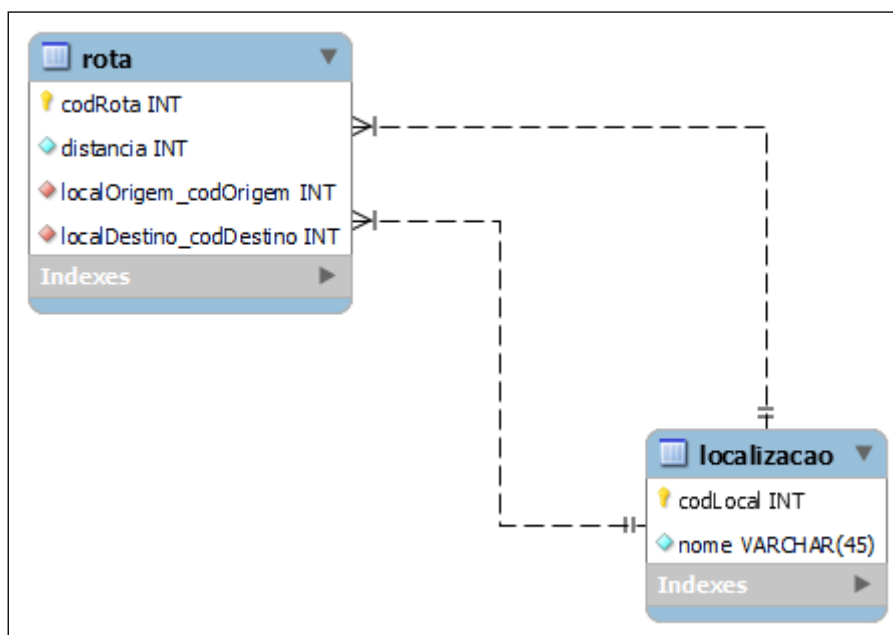


Figura 5. Modelo relacional de um banco de rotas de transporte

Onde temos a tabela localizacao que armazena os dados do local onde se inicia e termina a rota de entrega respectivamente. Na tabela rota, que é a tabela central, onde é registrada a rota, armazenando a distância entre dois locais, de origem até o destino e a referência para a tabela localizacao, onde estão cadastrados os dados do local de partida e de chegada.

Só que no banco temos registros de rotas onde origem e destino já estão registrados, mas que se diferenciam pelo caminho, fazendo com que a distância seja maior ou menor ou até igual.

Buscando sempre reduzir o custo de transporte o gerente da transportadora deseja que para rotas onde existe mais de uma alternativa entre o local de origem até o destino seja utilizado a rota que tem a menor distância.

Para saber qual é o menor caminho entre rotas onde origem e destino existem mais de um caminho será necessário utilizar uma linguagem para consultar essas informações no banco de dados. No modelo relacional é utilizado a linguagem SQL¹.

Segundo OLIVEIRA (2002, p.18), SQL é um conjunto de comandos de manipulação de banco de dados utilizado para criar e manter a estrutura desse banco de dados, além de incluir, excluir, modificar e pesquisar informações nas tabelas dele.

O autor também define que em um modelo relacional, apenas um tipo de estrutura de dados existe: a tabela. Novas tabelas são criadas com junção ou combinação de outras tabelas. Utilizando apenas um comando SQL é possível pesquisar dados em diversas tabelas ou atualizar ou excluir diversas linhas ou tabelas.

Exemplificando o uso da linguagem SQL para fazer as buscas pelas informações no banco de dados criado para o teste vamos demonstrar por meio dos dados que existem no banco de acordo com a figura 6 abaixo.

localizacao		localizacao	
codLocal	nome	codLocal	nome
1	A1	7	B1
2	A2	8	B2
3	A3	9	B3
4	A4	10	B4
5	A5	11	B5
6	A6	12	B6

rota			
codRota	distancia	localOrigem_codOrigem	localDestino_codDestino
1	100	1	12
2	30	2	9
3	20	3	10
4	110	1	12
5	40	2	10
6	200	1	7
7	90	1	12
8	80	2	8
9	80	3	11
10	50	6	12

Figura 6. Dados do banco de dados

Temos os dados dos registros da tabela localizacao. E na tabela de rotas estão as rotas cadastradas com código, distância e origem e destino da rota. Armazenando os códigos com referência aos registros que estão na outra tabela.

¹ SQL (Structured Query Language) que a tradução é Linguagem de Consulta Estruturada.

Por exemplo: a primeira linha de registro da tabela rota temos o código da rota igual 1, para uma distância 100 e com o local de origem 1 e local de destino 12. Se observar na tabela localizacao para o código origem 1 a origem é A1, já na tabela localizacao para o código destino 12 o destino é B6. Então para a rota 1 temos 100 de distância entre A1 e B6.

Voltando para a situação em que temos origem e destino que existem mais de uma rota é possível observar através da figura 7, onde as linhas estão destacadas em verde são origem e destino que possuem mais de uma alternativa para realizar o transporte.

rota			
codRota	distancia	localOrigem_codOrigem	localDestino_codDestino
1	100	1	12
2	30	2	9
3	20	3	10
4	110	1	12
5	40	2	10
6	200	1	7
7	90	1	12
8	80	2	8
9	80	3	11
10	50	6	12

Figura 7. Dados da tabela de rota

Para a origem A1 e o destino B6 temos 3 rotas onde o transporte pode ser realizado, cada um com uma distância diferente, 100, 110 e 90.

Fazendo uso da linguagem SQL, além de percorrer as 3 tabelas buscando por origem e destino que possuem mais de uma rota, temos que calcular qual das 3 rotas possui a menor distância, fazendo o uso de junções entre tabelas para chegar ao resultado. Após isso chegamos à conclusão que para transporte de origem A1 e destino B6 a rota 7 que possui distância igual a 90 é a melhor alternativa para realizar o transporte, visando o custo, por ter a menor distância entre as opções comparadas.

Em situações como essa, onde temos poucos registros e apenas uma origem e destino com mais de uma opção para fazer o percurso, é possível até ser tranquilo fazer as buscas, mas em bases maiores e com muitos registros ela pode se tornar pouco eficiente.

Usando o mesmo exemplo utilizado para a empresa de transporte, modelar o mesmo banco de dados só que orientado a Grafo, é possível ver como esse modelo trabalha para lidar com o problema de achar a rota que possui a menor distância entre as rotas para a mesma origem e destino.

Na figura 8, temos o nosso banco de dados modelado em forma de um Grafo.

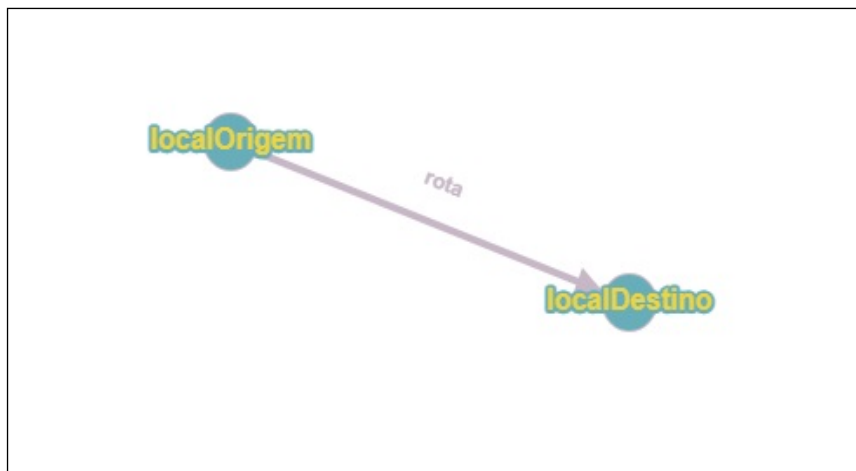


Figura 8. Modelo grafo de um banco de rotas de transporte

Onde o vértice localOrigem esse conecta-se, através de uma aresta rota, a outro vértice chamado localDestino.

Diferente do modelo relacional, o modelo Grafo não trabalha com a linguagem SQL. No orientado a Grafo se trabalha com algoritmos de Grafo. Temos diversas opções de algoritmos em grafos como: Dijkstra, Kruskal, Prim e entre outros.

Para usar como solução para o nosso problema para encontrar a rota que possui a menor distância vamos usar o algoritmo de Dijkstra.

Segundo GOLDBARG (2012, p. 200) Dijkstra (1995) apresentou um algoritmo de rotulação de vértices para encontrar os caminhos mais curtos de uma origem para os demais vértices em grafos com custos positivos nos arcos.

Então é possível começar projetando os dados do nosso banco (figura 6) em forma de grafo, como pode ser visto na figura 9 abaixo.

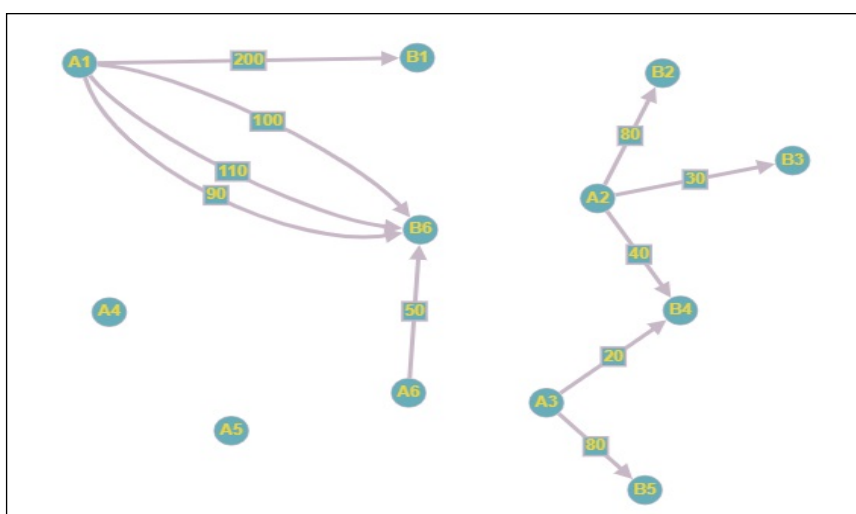


Figura 9. Grafo dos dados no banco de transportes

No grafo, projeto do registro do banco de dados de rotas de transportes, tem os vértices que representam os pontos de origem e destino e as arestas que ligam a origem, de onde

se inicia o transporte, até o destino final. Na aresta temos um dado importante, que é a distância entre origem e destino. Com isso conseguimos ter uma informação importante em relação à representação da rota através do Grafo, que através de um vértice de origem ligado por uma aresta de valor x e que representa a nossa rota chegamos a um vértice de destino.

Com base nesse grafo onde da origem A1 e destino B6 existem 3 opções de rotas para realizar o transporte, onde cada rota possui 90, 110 e 110 de distância de origem ao destino final. Aplicando o algoritmo de Dijkstra, ele vai calcular a distância do vértice A1 até o vértice B6, passando por todos os caminhos possíveis até encontrar o menor caminho e com isso ele vai chegar à conclusão de que do vértice A1 até o B6 pela rota de distância 90 é o melhor caminho a ser fazer, levando em conta a menor distância para diminuir o custo do transporte.

Nessa situação em que usando um problema de transporte e aplicando diferentes soluções usando o modelo relacional e o orientado a Grafo pode-se ver que, dependendo do problema, mesmo a solução estando em um modelo relacional ela pode ter uma adequação melhor no modelo Grafo.

Foi possível observar que no relacional a junção de tabelas foi um problema quando se teve que calcular qual menor rota entre rotas que origem e destino são iguais, pois qual seria a consulta SQL adequada para se encontrar o menor caminho entre uma origem e destino?

Já na representação em Grafo os dados se adéquo de uma melhor forma a estrutura, trazendo mais sentido e informação e sendo uma solução menos custosa ao se realizar a busca pelo caminho de menor distância por já existir uma solução pronto para trazer esses problemas, que é o algoritmo de Dijkstra.

Isso não significa que todos os modelos em relacional teriam o mesmo resultado ao serem projetados em modelos grafos, pois a solução para um problema não significa que serve para todos os outros problemas.

5. Proposta para conversão entre os modelos

Tendo em vista que o modelo relacional de banco de dados é amplamente adoado e maduro. Exergou se então a necessidade de propor a conversão de modelos de banco de dados existentes e que foram modelados no relacional para serem convertidos para o modelo Grafo, com isso mapear a estrutura de um banco relacional que é formado por tabelas e relacionamentos em um Grafo que é composto por vértices e arestas. Lembrando que isso não significa que todo modelo relacional deve ser convertido para Grafo e que o modelo relacional não é recomendado.

Para modelar o banco de dados relacional foi utilizada a ferramenta de case DBDesigner que serve para fazer a modelagem visual de dados.

Na ferramenta DBDesigner foi modelado um banco de dados para registrar a leitura de livros feita por leitores de uma biblioteca, que pode ser visualizada na figura abaixo.

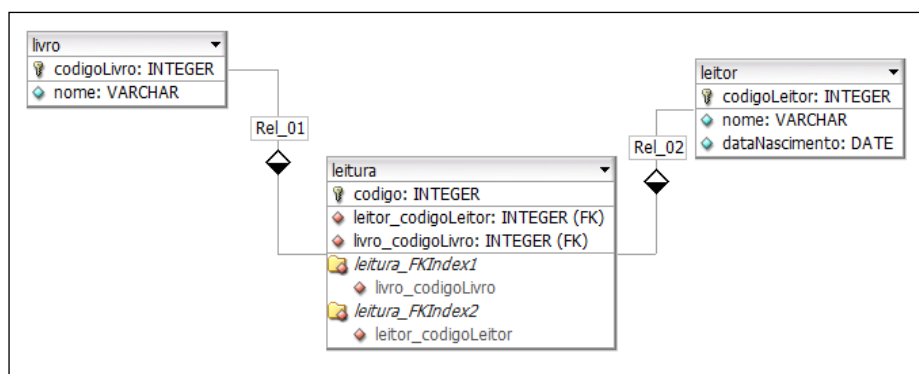


Figura 10. Modelagem relacional de banco de dados de registro de leituras

Nessa modelagem temos a tabela livro e leitor e uma terceira tabela que se relaciona com as tabelas livro e leitor, registrando todas as leituras de livros feitas pelos leitores.

Na própria ferramenta DBDesigner é possível salvar a modelagem em um formato XML.

Após salvar a modelagem no formato XML foi feito o desenvolvimento de uma ferramenta em linguagem Java, para fazer a leitura do arquivo XML em que está a modelagem do banco de dados.

Já com a ferramenta fazendo a leitura do arquivo, a mesma também transformou os objetos do modelo relacional, tabelas e relacionamentos em objetos do modelo Grafo, vértices e arestas respectivamente.

Com a ferramenta já fazendo a conversão de relacional para grafo, foi feita a persistência do modelo Grafo em um arquivo de formato .graphml.

Graphml é um formato de arquivo baseado no XML para Grafos.

Com o modelo Grafo salvo em um formato .graphml foi possível projetar o Grafo de formato visual através de uma ferramenta que faz a leitura desse tipo de arquivo. Como pode ser visto na figura abaixo.

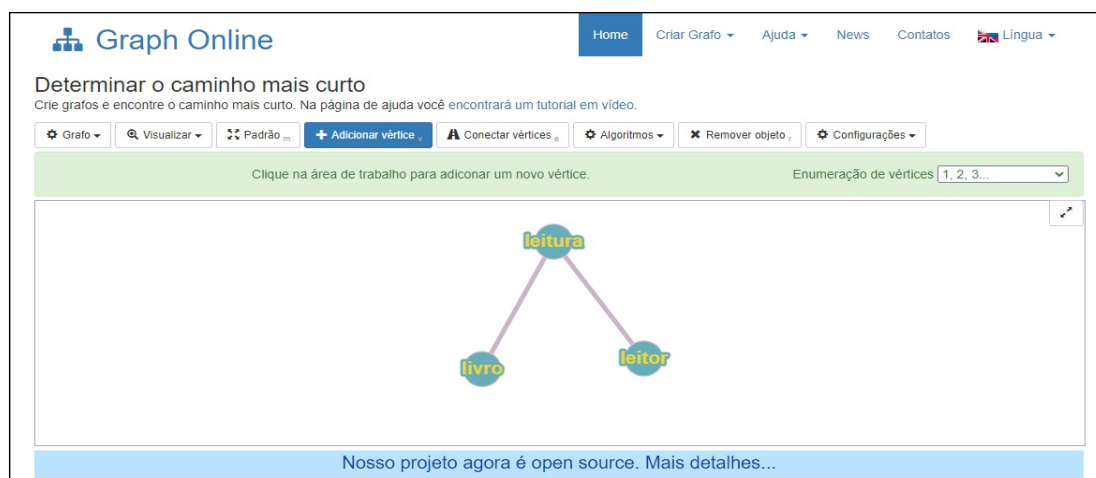
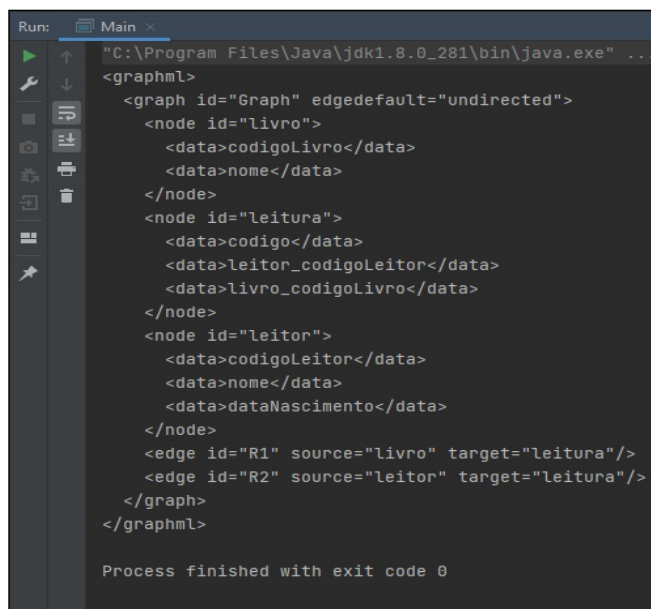


Figura 11. Grafo projetado em uma ferramenta online

6. Resultados Obtidos

Ao fazer essa abordagem de transformar as tabelas em vértices e os relacionamentos em arestas, que é possível observar na figura abaixo. Que é o resultado do mapeamento do modelo relacional para o Grafo feito pela ferramenta desenvolvida.



```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
<graphml>
<graph id="Graph" edgedefault="undirected">
  <node id="livro">
    <data>codigoLivro</data>
    <data>nome</data>
  </node>
  <node id="leitura">
    <data>codigo</data>
    <data>leitor_codigoLeitor</data>
    <data>livro_codigoLivro</data>
  </node>
  <node id="leitor">
    <data>codigoLeitor</data>
    <data>nome</data>
    <data>dataNascimento</data>
  </node>
  <edge id="R1" source="livro" target="leitura"/>
  <edge id="R2" source="leitor" target="leitura"/>
</graph>
</graphml>

Process finished with exit code 0
```

Figura 12. Conversão para modelo grafo feito pela ferramenta desenvolvida

As tabelas foram mapeadas para as tags <node> sendo identificado por um identificador que é o nome do vértice, que antes era uma tabela no relacional. Dentro das tags de cada vértice temos um conjunto de tags <data> que representa os campos dos vértices.

A tag <edge> representa as arestas, que veio do modelo relacional como um relacionamento entre tabelas.

Com as arestas também com identificadores únicos e com atributos que definem qual vértice está ligado a qual vértice.

Analisando visualmente o Grafo gerado, figura 13.

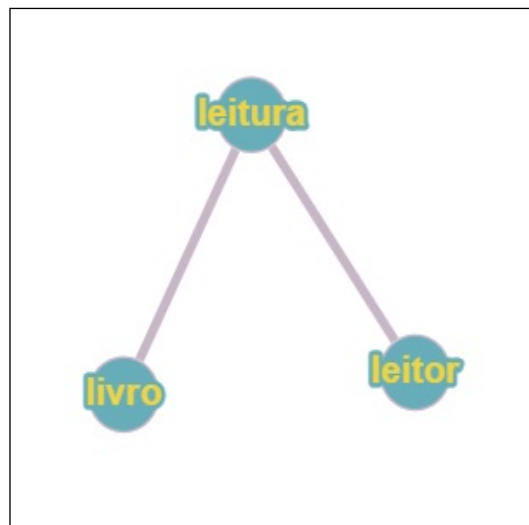


Figura 13. Modelo grafo gerado pela ferramenta a partir do modelo relacional

Suponhamos que temos o leitor Pedro e o livro Dicas e queremos informar nesse modelo que Pedro fez a leitura do livro Dicas.

Seria necessário utilizar 3 vértices e 2 arestas para representar essa informação.

Com o modelo relacional sendo mapeado com a ideia de transformar tabela em vértice e relacionamento em uma aresta, passa a ideia de estarmos migrando os modelos apenas para representações diferentes e mantendo a mesma complexidade para ser lidar com a informação.

7. Considerações Finais

No trabalho foi discutido sobre dois modelos de banco de dados, o relacional e o orientado a grafo.

Devido a grande importância que a ferramenta de banco de dados tem para as organizações é importantíssimo estudar modelos em que a abordagem seja a mais compatível possível para a resolução dos problemas.

Estudando os modelos em problemas e apresentando discussões sobre como funciona cada abordagem ao se aplicar as soluções, foi possível constatar que fazer migrações de modelos relacionais para orientado a grafo não é uma garantia de melhores resultados.

Ao cogitar fazer migrações do relacional para o grafo deve se levar em consideração para que tipo de problema de negócio o banco de dados vai trabalhar.

Para situações onde o modelo entidade relacionamento possui soluções apropriadas para lidar com o problema não é recomendável fazer esses tipos de migrações.

A não ser que o modelo grafo tenha soluções melhores e que traga vantagens, aí sim essa migração pode ser viável.

Através da proposta de conversão de modelo relacional para modelo grafo, foi possível identificar que essa ideia de simplesmente pegar um modelo relacional e levar para o grafo, não funciona. Ainda mais quando é feito um mapeamento direto de tabela para vértice e relacionamento para aresta.

Muitas vezes isso faz com que se mude apenas o modelo e levando consigo a sua complexidade.

Com isso é possível entender que o que pode funcionar em um modelo, não significa que vai funcionar para o outro.

Referências

- FOWLER, M.; SADALAGE, P. J. NoSQL Distilled. [S.l.]: Addison-Wesley, 2012. ISBN 978-0-387-79710-6.
- MATSUMOTO, C. Y. A IMPORTÂNCIA DO BANCO DE DADOS EM UMA ORGANIZAÇÃO. *Maringá Management: Revista de Ciências Empresariais*, Maringá - PR, v.3, n.1, p.45-55, janeiro/2006.
- MACHADO, F. N. R.; ABREU, M. P. de. Projeto de banco de dados: uma visão prática. 1ª ed. São Paulo: Érica, 1996. p14.
- EICH, M. H.; MISHRA, P. Join processing in relational databases. *ACM Computing Surveys*, v. 24, n. 1, p. 1–51, 1992. Disponível em: <<https://www.csd.uoc.gr/~hy460/pdf/p63-mishra.pdf>>. Citado na página 21.
- FACHINELLI, T. P. Análise de Projetos de Banco de Dados: Modelo Relacional vs. Modelo em Grafos. 2019. 66. (Ciência Exatas e Aplicadas) – Universidade Federal de Ouro Preto, João Monlevade–MG, 2019.
- ALVAREZ, G. M; CECI. F. Base de Dados Orientada a Grafos: Um Experimento aplicado na Análise Social. *Revista Tecnologia e Sociedade*, Curitiba, v.11. n.21, 127-139, janeiro/2015.
- FILHO. S. P. L; CAVALCANTI. M. C; JUSTEL. C. M. Modelagem em Grafos a partir de Bancos de Dados Relacionais. Rio de Janeiro, setembro/2017.
- SOUSA, Victor Martins. Projeto Lógico de Banco de Dados NoSQL de Grafos a partir de um modelo conceitual baseado no modelo Entidade-Relacionamento. 2018. 111. (Ciências exatas) – UNIFACCAMP, Campo Limpo Paulista, 2018.
- OLIVEIRA, A. Exemplo de aplicações com INNER JOIN. *Blogson*, 2020. Disponível em: <https://www.blogson.com.br/exemplo-de-aplicacoes-com-join/>. Acesso em: 02/02/2022.
- GOLDBARG, M; GOLDBARG, E. Grafos: Conceitos, algoritmos e aplicações. 2ª ed. Rio de Janeiro: Elsevier, 2012.
- OLIVEIRA, C. H. P. SQL: Curso Prático. 1. ed. São Paulo: Novatec Editora, 2002. Knuth, D. E. (1984), *The TeXbook*, Addison Wesley, 15th edition.
- MEDEIROS, H. DBDesigner: Modelagem e Implementação de banco de dados. *DEV MEDIA*, 2014. Disponível em: <https://www.devmedia.com.br/dbdesigner-modelagem-e-implementacao-de-banco-de-dados/30897>. Acesso em: 02/02/2022.