

Trabalho Final DB - Neo4j

Adicionando categorias de produtos

```
MERGE (:Categoria {nome: "Eletrônicos"})
MERGE (:Categoria {nome: "Calçados"})
MERGE (:Categoria {nome: "Acessório"})
MERGE (:Categoria {nome: "Roupa"})
MERGE (:Categoria {nome: "Informática"})
```

Adicionando produtos com relacionamento com categorias

```
MERGE (p1:Produto {id: "p101", nome: "Tênis Esportivo"})-[:PERTENCE_A]→(:Categoria {nome: "Eletrônicos"})
MERGE (p2:Produto {id: "p102", nome: "Camiseta Dry Fit"})-[:PERTENCE_A]→(:Categoria {nome: "Roupa"})
MERGE (p3:Produto {id: "p201", nome: "Notebook Gamer"})-[:PERTENCE_A]→(:Categoria {nome: "Informática"})
MERGE (p4:Produto {id: "p202", nome: "Mouse Sem Fio"})-[:PERTENCE_A]→(:Categoria {nome: "Eletrônicos"})
MERGE (p5:Produto {id: "p203", nome: "Cabo USB-C"})-[:PERTENCE_A]→(:Categoria {nome: "Eletrônicos"})
MERGE (p6:Produto {id: "p301", nome: "Smartphone"})-[:PERTENCE_A]→(:Categoria {nome: "Eletrônicos"})
MERGE (p7:Produto {id: "p302", nome: "Headset Gamer"})-[:PERTENCE_A]→(:Categoria {nome: "Eletrônicos"})
```

Criando relacionamento entre Categorias "Eletrônicos" e "Informática"

```
MATCH (a:Categoria {nome: "Eletrônicos"}), (b:Categoria {nome: "Informática"})
MERGE (a)-[:RELACIONADA_COM]→(b)
```

Adicionando clientes

```
MERGE (c1:Cliente {id: 201, nome: "Carla"})
MERGE (c2:Cliente {id: 202, nome: "Pedro"})
MERGE (c3:Cliente {id: 203, nome: "Lucas"})
MERGE (c4:Cliente {id: 204, nome: "Marcos"})
```

Visualização de categorias

```
MATCH (c:Cliente), (cat:Categoria)
MERGE (c)-[:VISUALIZOU_CATEGORIA]->(cat)
```

Todos os clientes visualizaram todas as categorias

Adicionando Pedidos com Produtos comprados

```
MERGE (c1:Cliente {id: 201})
MERGE (pA:Pedido {id: "ped1"}) <-[:REALIZOU]-(c1)
MERGE (pA)-[:CONTEM]->(:Produto {id: "p201"})
MERGE (pA)-[:CONTEM]->(:Produto {id: "p301"})

MERGE (c2:Cliente {id: 202})
MERGE (pB:Pedido {id: "ped2"}) <-[:REALIZOU]-(c2)
MERGE (pB)-[:CONTEM]->(:Produto {id: "p201"})
MERGE (pB)-[:CONTEM]->(:Produto {id: "p302"})

MERGE (c3:Cliente {id: 203})
MERGE (pC:Pedido {id: "ped3"}) <-[:REALIZOU]-(c3)
MERGE (pC)-[:CONTEM]->(:Produto {id: "p101"})

MERGE (c4:Cliente {id: 204})
MERGE (pD:Pedido {id: "ped4"}) <-[:REALIZOU]-(c4)
MERGE (pD)-[:CONTEM]->(:Produto {id: "p301"})
MERGE (pD)-[:CONTEM]->(:Produto {id: "p202"})
```

```
MERGE (pE:Pedido {id: "ped5"})←[:REALIZOU]-(c4)
MERGE (pE)-[:CONTEM]→(:Produto {id: "p301"})
```

Adicionando avaliações de clientes

```
MATCH (c1:Cliente {id: 201}), (p1:Produto {id: "p201"})
MERGE (c1)-[:AVALIOU {nota: 5, comentario: "Ótimo notebook"}]→(p1)
WITH 1 AS dummy
```

```
MATCH (c2:Cliente {id: 202}), (p2:Produto {id: "p302"})
MERGE (c2)-[:AVALIOU {nota: 4, comentario: "Bom notebook, headset razoáv"}]→(p2)
WITH 1 AS dummy
```

```
MATCH (c3:Cliente {id: 203}), (p3:Produto {id: "p101"})
MERGE (c3)-[:AVALIOU {nota: 5, comentario: "Confortável e leve"}]→(p3)
WITH 1 AS dummy
```

```
MATCH (c4:Cliente {id: 204}), (p4:Produto {id: "p202"})
MERGE (c4)-[:AVALIOU {nota: 4, comentario: "Mouse muito preciso"}]→(p4)
WITH 1 AS dummy
```

```
MATCH (c4:Cliente {id: 204}), (p5:Produto {id: "p301"})
MERGE (c4)-[:AVALIOU {nota: 5, comentario: "Smartphone excelente"}]→(p5)
```

Filtragem Colaborativa (Item-Item)

```
MATCH (:Produto {id: "p201"})←[:CONTEM]-(pedido:Pedido)-[:CONTEM]→(outro:Produto)
WHERE outro.id <> "p201"
RETURN outro.id AS produto_relacionado, COUNT(*) AS vezes_comprado_juntos
ORDER BY vezes_comprado_juntos DESC
```

Faz uma busca dos produtos que foram comprados juntamente com o produto 'p201'.

Filtragem Colaborativa (User-User)

```
MATCH (c:Cliente {id: 201})-[:REALIZOU]→(:Pedido)-[:CONTEM]→(p:Produto)
WITH c, COLLECT(DISTINCT p) AS produtos_cliente
```

```
MATCH (outro:Cliente)-[:REALIZOU]→(:Pedido)-[:CONTEM]→(p_comum:Produto)
WHERE p_comum IN produtos_cliente AND outro <> c
WITH c, produtos_cliente, outro
```

```
MATCH (outro)-[:REALIZOU]→(:Pedido)-[:CONTEM]→(recomendado:Produto)
WHERE NOT recomendado IN produtos_cliente
RETURN DISTINCT recomendado.id AS id_produto, recomendado.nome AS nome_produto
ORDER BY nome_produto
```

Consulta clientes que compraram os mesmos produtos de um cliente X, no caso acima, o cliente 201, e retorna os produtos que o cliente em questão não comprou mas que foram comprados pelos clientes relacionados.

Identificando caminho mais curto entre dois produtos através de suas categorias.

```
MATCH (p1:Produto {id: "p201"})-[:PERTENCE_A]→(c1:Categoria),
      (p2:Produto {id: "p302"})-[:PERTENCE_A]→(c2:Categoria),
      path = shortestPath((c1)-[:RELACIONADA_COM*]-(c2))
RETURN path
```

Mostra o caminho mais curto entre os produtos p201 e p302, no caso, as categorias relacionadas anteriormente.

Identificando clientes influenciadores (cujas avaliações positivas se correlacionam com mais vendas).

```
MATCH (c:Cliente)-[a:AVALIOU]→(p:Produto)
WHERE a.nota >= 4
```

```

MATCH (p)←[:CONTEM]-(:Pedido)←[:REALIZOU]-(outro:Cliente)
WHERE outro <> c

RETURN c.nome AS cliente_influenciador,
       p.nome AS produto_avaliado,
       COUNT(DISTINCT outro) AS clientes_impactados,
       COUNT(*) AS total_vendas_pos_avaliacao
ORDER BY clientes_impactados DESC, total_vendas_pos_avaliacao DESC

```

Identifica os clientes que avaliaram um produto com a nota maior ou igual a 4 e que foram comprados por outros clientes posteriormente.

Recomendendo produtos de categorias visualizadas, mas não compradas

```

MATCH (c:Cliente {id: 201})-[:REALIZOU]→(:Pedido)-[:CONTEM]→(p:Produto)
WITH c, COLLECT(DISTINCT p.id) AS produtos_comprados

MATCH (p:Produto)-[:PERTENCE_A]→(cat1:Categoria)
WHERE p.id IN produtos_comprados
WITH c, produtos_comprados, COLLECT(DISTINCT cat1.nome) AS categorias_

MATCH (c)-[:VISUALIZOU_CATEGORIA]→(cat2:Categoria)
WITH categorias_compradas, COLLECT(DISTINCT cat2.nome) AS categorias_

WITH [cat IN categorias_visualizadas WHERE NOT cat IN categorias_compradas]

MATCH (p:Produto)-[:PERTENCE_A]→(cat:Categoria)
WHERE cat.nome IN categorias_recomendadas
RETURN DISTINCT p.id AS id_produto, p.nome AS nome_produto, cat.nome AS categoria
ORDER BY categoria, nome_produto

```

Faz uma busca para o cliente 201, onde compara as categorias visualizadas com as categorias dos produtos comprados e retorna a lista de produtos que foram visualizados e não foram comprados.