TEMPLATE - POTENCIALIZANDO O DESEMPENHO COM NOSQL

Carlos Vinícius Rodrigues Silva - RM564607
Gabriela Sena da Silva - RM565118
Gustavo Almeira Scardini - RM 565374
Tatiana Espinola - RM 564907
Vitor Fernandes Antunes - RM 563053

SUMÁRIO

1. PROVA DE CONCEITO DE BANCO DE DADOS NOSQL

1.1 Análise de cenários

A realização de testes de cenários é importante porque permite validar, na prática, se a solução proposta realmente atende às necessidades do negócio, evitando que a escolha fique apenas no campo teórico. Esses testes reduzem riscos e custos, já que identificam problemas e limitações técnicas antes da implementação definitiva. Além disso, possibilitam avaliar desempenho, escalabilidade e aderência da tecnologia às demandas estratégicas da empresa, fornecendo evidências concretas para apoiar a tomada de decisão.

1.2 Cenário 1

Quando um cliente seleciona um produto, a plataforma de e-commerce exibe, adicionalmente, recomendações de outros itens, baseadas nas compras de quem comprou esse produto e em outras promoções correlatas. No contexto atual, esse cálculo está demorando muito tempo para ser feito utilizando estruturas relacionais, dado o volume de dados envolvidos.

Para esse primeiro cenário, a TI propôs o uso de um banco de dados NoSQL do tipo GRAFO.

1.3 Justificativa do cenário 1

O uso de um banco do tipo grafo é adequado para este cenário porque o problema envolve relações complexas entre entidades. Bancos de dados de grafos permitem armazenar entidades (clientes, produtos) como nós e relacionamentos (quem comprou o quê, produtos correlatos) como arestas, possibilitando consultas eficientes focadas nos relacionamentos, como recomendações baseadas em produtos comprados juntos. Diferentemente de um banco relacional, que exigiria múltiplos joins em grandes volumes de dados, o grafo facilita a navegação e recuperação de padrões complexos de maneira muito mais eficiente.

1.3.1 Empresa que usa o cenário 1

Facebook utiliza bancos de dados de grafos (Neo4j) para gerenciar conexões entre usuários, amigos e interesses, otimizando consultas baseadas em relacionamentos.

1.4 Cenário 2

A definição da entrega de um produto em 24h depende da disponibilidade de estoque do centro de distribuição mais próximo do endereço de entrega. Se o cliente optar por essa entrega fast, é necessário realizar a reserva no centro de distribuição e, automaticamente, atualizar o estoque para atender a outros clientes. Nos testes preliminares com o uso do modelo relacional, o desempenho foi frustrante, influenciado principalmente pelo volume de dados e frequência de atualização.

Para esse segundo cenário, a TI propôs o uso de um banco de dados NoSQL do tipo COLUNAR.

1.4.1 Justificativa do cenário 2

Para a gestão do estoque um banco colunar é bastante viável, pois são otimizadas para consultas e leituras rápidas em colunas específicas, mesmo em grandes volumes de dados. Esse tipo de banco permite atualizar ou consultar rapidamente a quantidade de produtos disponíveis em centros de distribuição próximos ao cliente, sem precisar ler linhas inteiras do banco, como ocorreria em um modelo relacional tradicional. A abordagem colunar oferece alto desempenho em agregações e consultas frequentes de inventário, sendo ideal para operações que exigem velocidade e consistência na leitura de dados, além de continuar funcionando caso ocorram instabilidades em algum dos clusters que rodam o banco.

1.4.2 Empresa que usa o cenário 2

Walmart utiliza bancos de dados colunares (Cassandra) para armazenar dados de estoque, atualizando inventário de lojas instantaneamente quando compras são realizadas.

1.5 Cenário 2

A tela de detalhes de um produto sempre recebe novas informações e, hoje em dia, possui informações que podem ser armazenadas juntamente com o produto, tais como: reviews do produto; suas versões; informações de entrega; imagens; recomendações; dicas, entre outras.

Para esse terceiro cenário, a TI propôs o uso de um banco de dados NoSQL do tipo DOCUMENTO.

1.5.1 Justificativa do cenário 3

O banco de dados NoSQL do tipo documento é apropriado para armazenar informações variadas e dinâmicas de produtos, como revisões, versões, imagens e dicas, pois permite armazenar documentos JSON completos, cada um representando um produto com todos os seus atributos. Essa flexibilidade evita a rigidez do modelo relacional, que exigiria múltiplas tabelas e joins para armazenar dados heterogêneos ou até mudanças na arquitetura do banco. Além disso, atualizações frequentes de atributos distintos de um mesmo produto podem ser feitas sem impactar outras entidades.

1.5.2 Empresa que usa o cenário 3

Sephora utiliza MongoDB para gerenciar seu catálogo de produtos, uma vez que nem todos os produtos possuem exatamente os mesmos dados a serem armazenados.

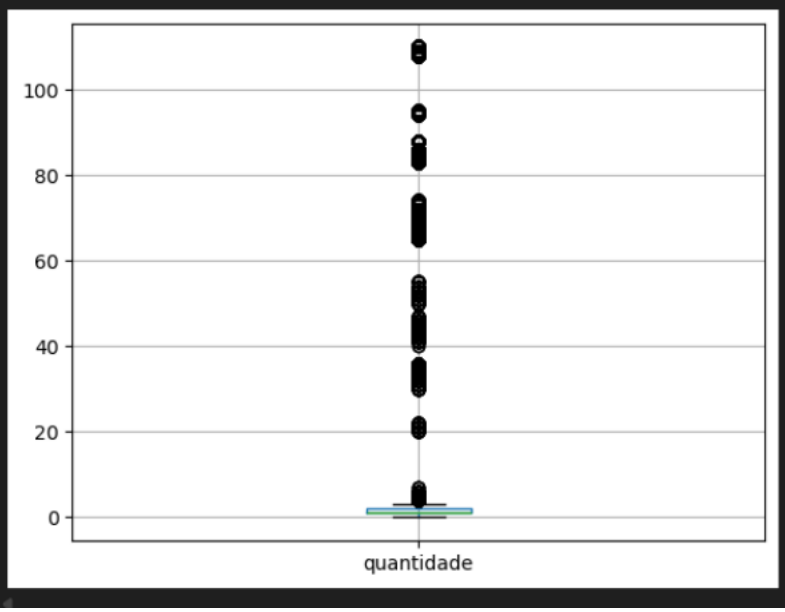
2. ANÁLISE DOS DADOS DE VENDAS

2.1 Quantidade

```
# Desvio padrão que 4x a média e valor máximo 55x o o valor do 3Q, indicativos de possíveis outliers

# Visualização dos dados de quantidade em um gráfixo box-plot para verificação da existência de outliers

vendas_q2[['quantidade']].boxplot()
plt.show()
```




```
# Outra maneira de classificar outliers (Z score)

vendas_q2['quantidade_z'] = scale(vendas_q2['quantidade'])

# -3 e +3 utilizados como limites de Z Score para configuração de outlier

outliers_z = vendas_q2[(vendas_q2['quantidade_z'] > 3) | (vendas_q2['quantidade_z'] < -3)]

outliers_z.sort_values('quantidade_z')
```

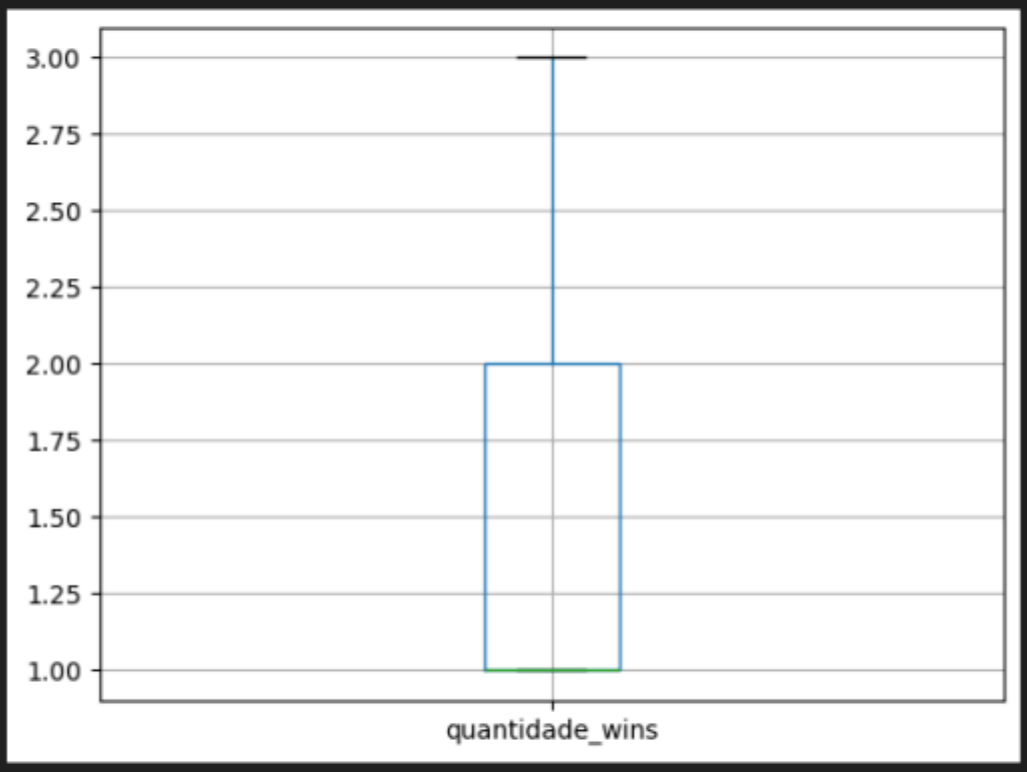
	cod_pedido	regiao_pais	produto	valor	quantidade	valor_total_bruto	data	estado	formapagto	centro_distribuicao	responsavelpedido	valor_comissao	lucro_liquido	categoriaprod
101815	102010	Norte	Biscoito True Champion 300g	27.93	40.0	1117.20	02/11/2021	Amazonas	Cartão Crédito	Rapid Pink	Silvia	16.80	336.00	Alimentação
150419	150614	Centro Oeste	Bola Pet Vinil Big Blue	30.78	40.0	1231.20	26/02/2022	Mato Grosso	Cartão Débito	Grãos Blue	Andressa	7.60	304.00	Brinquedo
194063	194258	Norte	Whiskas Petisco Temptations Anti Bola de Pelo 40g	11.34	41.0	464.94	27/02/2022	Pará	Boleto Bancário	Rapid Pink	Julia	6.15	61.50	Petisco
199896	200091	Centro Oeste	Kit Banho e Tosa com Escova PetShop Cãopeon	103.68	41.0	4250.88	07/04/2022	Distrito Federal	Cartão Débito	Grãos Blue	Lucia	129.15	2066.40	Higiene e Limpeza
199820	200015	Centro Oeste	Suplemento Alimentar Glutamina Mundo Animal Nu...	59.94	41.0	2457.54	01/03/2022	Mato Grosso do Sul	Dinheiro	Grãos Blue	Ligia	40.59	744.15	Medicamento

```
# Visualização dos dados winsorizados para checar se ainda existem outliers

vendas_q2[['quantidade_wins']].boxplot()
plt.show()

# Visualização de estatísticas básicas após tratamento de outliers

vendas_q2['quantidade_wins'].describe()
```




```
count    200119.000000
mean          1.658088
std           0.821446
min           1.000000
25%           1.000000
50%           1.000000
75%           2.000000
max           3.000000
Name: quantidade_wins, dtype: float64
```

Resposta da pergunta: Sim, existem outliers nos dados da coluna 'quantidade'. Sim, é possível notar que essas vendas são de valores bastante elevados e de grande quantidade de produtos, provavelmente sendo compras realizadas por outras empresas. Sem a winsorização, encontramos um desvio padrão próximo de 12 para a coluna quantidade. Após a winsorização e consequente tratamento de todos os possíveis outliers (valor máximo deixou de ser 110 e passou a ser 3), o novo desvio padrão é próximo de 0,82.

2.2 Preço

```
# Cálculo da média de preço da população geral

media_geral = vendas_q3['valor'].mean().round(2)
total_registros = len(vendas_q3)

media_geral
```

```
111.09
```

```
# Cálculo da média de preço por região

media_por_regiao = vendas_q3.groupby('regiao_pais').agg(Media_Valor=('valor', 'mean'), Quantidade=('valor', 'count')).reset_index()
media_por_regiao['Media_Valor'] = media_por_regiao['Media_Valor'].round(2)

media_por_regiao
```

	regiao_pais	Media_Valor	Quantidade
0	Centro Oeste	111.23	32121
1	Nordeste	111.10	48205
2	Norte	111.10	56198
3	Sudeste	111.08	40167
4	Sul	110.87	24095

```
# Cálculo da média de preço por forma de pagamento

media_por_formapagto = vendas_q3.groupby('formapagto').agg(Media_Valor=('valor', 'mean'), Quantidade=('valor', 'count')).reset_index()
media_por_formapagto['Media_Valor'] = media_por_formapagto['Media_Valor'].round(2)

media_por_formapagto
```

	formapagto	Media_Valor	Quantidade
0	Boleto Bancário	110.59	40281
1	Cartão Crédito	112.51	40507
2	Cartão Débito	109.63	40032
3	Dinheiro	111.48	39792
4	Pix	111.23	40174

Resposta da pergunta: Não houve diferença estatisticamente relevante entre a média de preço entre a média geral da população e a média de qualquer região ou qualquer forma de pagamento. Os valores mais distantes da média geral foram os de compras realizadas por cartão (crédito 1,02% maior e débito 1,31% menor), nenhum configurando uma diferença relevante

2.3 Correlações

```
# Cálculo da matriz de correlação das variáveis numéricas

pearson = vendas_q4[['valor', 'quantidade', 'valor_total_bruto', 'valor_comissao', 'lucro_liquido']].corr()

# matriz ajustada
pearson.style.background_gradient(cmap='coolwarm').format("{:.2f}")
```

	valor	quantidade	valor_total_bruto	valor_comissao	lucro_liquido
valor	1.00	-0.01	0.06	0.21	0.19
quantidade	-0.01	1.00	0.12	0.32	0.56
valor_total_bruto	0.06	0.12	1.00	0.26	0.26
valor_comissao	0.21	0.32	0.26	1.00	0.89
lucro_liquido	0.19	0.56	0.26	0.89	1.00

Resposta da pergunta: Não há nenhuma forte correlação negativa em nossa matriz de correlação, porém existem duas fortes positivas, uma muito forte inclusive. As colunas quantidade e lucro_liquido apresentam um índice de correlação de 0.56 (forte) e as colunas valor_comissao e lucro_liquido 0.89 (muito forte).