

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL - ESCOLA
POLITÉCNICA

GABRIEL DORNELES REBELLO
TURMA 13

APRESENTAÇÃO DAS QUERIES: COMPARAÇÃO ENTRE SQL E NoSQL

Porto Alegre
2024

GRADUAÇÃO



Pontifícia Universidade Católica
do Rio Grande do Sul

1: Consultar quantos pedidos pendentes a transportadora tem

SQL:

```
SELECT
    t.nome AS Nome_da_transportadora,
    p.id_pedido AS Id_do_pedido,
    pr.nome AS Nome_do_produto,
    pp.quantidade AS Quantidade_de_itens,
    pp.data_entrega AS Data_de_entrega,
    pp.status_entrega AS Status_da_entrega
FROM Transportadora t
INNER JOIN Pedido_has_produto pp ON t.id_transportadora = pp.id_transportadora
INNER JOIN Pedido p ON pp.id_pedido = p.id_pedido
INNER JOIN Produto pr ON pp.id_produto = pr.id_produto
WHERE pp.status_entrega = 'Pendente';
```

NOME_DA_TRANSPORTADORA	ID_DO_PEDIDO	NOME_DO_PRODUTO	QUANTIDADE_DE_ITENS	DATA_DE_ENTREGA	STATUS_DA_ENTREGA
Sedex	2	O Pequeno Príncipe	2	05-SEP-24	Pendente
Sedex	2	Ayrton - O herói Revelado	1	05-SEP-24	Pendente
Sedex	5	Diário de um Banana: Dias de Cão	2	12-SEP-24	Pendente
Correios	7	Diário de um Banana: Dias de Cão	1	05-JUN-24	Pendente
Sedex	5	Crepúsculo	1	12-SEP-24	Pendente

NoSQL:

```
# 1: Consultar quantos pedidos pendentes a transportadora tem

query = pedidos.find({
    "produtos.status_entrega": "Pendente"
})

resultados = []

for pedido in query:
    for produto in pedido['produtos']:
        if produto['status_entrega'] == 'Pendente':
            transportadora = transportadoras.find_one({
                "_id": produto['transportadora']['_id']
            })

            produto_info = produtos.find_one({
                "_id": produto['_id']
            })

            resultado = {
                "Nome_da_transportadora": transportadora['nome'],
                "Id_do_pedido": pedido['_id'],
                "Nome_do_produto": produto_info['nome'],
                "Quantidade_de_itens": produto['quantidade'],
                "Data_de_entrega": produto['data_entrega'],
                "Status_da_entrega": produto['status_entrega']
            }
            resultados.append(resultado)

for resultado in resultados:
    print(resultado)

# Explicacao: de inicio colocamos na 'query' todos os pedidos que tem 'status_entrega' em 'pendente', para entao trabalharmos com esses dados a fim de conseguir selecionar as informacoes gerais do pedido e poder buscar...
# a transportadora do pedido, entao colocando os dados selecionados na variavel 'resultado' e colocando-a em 'resultados' que sera percorrido e impresso com os dados buscados.

{'Nome_da_transportadora': 'Sedex', 'Id_do_pedido': 2, 'Nome_do_produto': 'O Pequeno Príncipe', 'Quantidade_de_itens': 2, 'Data_de_entrega': datetime.datetime(2024, 9, 5, 0, 0), 'Status_da_entrega': 'Pendente'}
{'Nome_da_transportadora': 'Sedex', 'Id_do_pedido': 2, 'Nome_do_produto': 'Ayrton - O herói Revelado', 'Quantidade_de_itens': 1, 'Data_de_entrega': datetime.datetime(2024, 9, 5, 0, 0), 'Status_da_entrega': 'Pendente'}
{'Nome_da_transportadora': 'Sedex', 'Id_do_pedido': 5, 'Nome_do_produto': 'Diário de um Banana: Dias de Cão', 'Quantidade_de_itens': 2, 'Data_de_entrega': datetime.datetime(2024, 9, 12, 0, 0), 'Status_da_entrega': 'Pendente'}
{'Nome_da_transportadora': 'Sedex', 'Id_do_pedido': 5, 'Nome_do_produto': 'Crepúsculo', 'Quantidade_de_itens': 1, 'Data_de_entrega': datetime.datetime(2024, 9, 12, 0, 0), 'Status_da_entrega': 'Pendente'}
{'Nome_da_transportadora': 'Correios', 'Id_do_pedido': 7, 'Nome_do_produto': 'Diário de um Banana: Dias de Cão', 'Quantidade_de_itens': 1, 'Data_de_entrega': datetime.datetime(2024, 6, 5, 0, 0), 'Status_da_entrega': 'Pendente'}
```

2: Consultar quais pessoas compraram produtos acima de mil reais e como foi pago

SQL:

```
SELECT
    c.nome AS Nome_do_cliente,
    pr.nome AS Nome_do_produto,
    pr.preco AS Preco_do_produto,
    p.forma_pgto AS Forma_de_pagamento
FROM Cliente c
INNER JOIN Pedido p ON c.id_cliente = p.id_cliente
INNER JOIN Pedido_has_produto pp ON p.id_pedido = pp.id_pedido
INNER JOIN Produto pr ON pp.id_produto = pr.id_produto
WHERE pr.preco > 1000.00;
```

NOME_DO_CLIENTE	NOME_DO_PRODUTO	PRECO_DO_PRODUTO	FORMA_DE_PAGAMENTO
Charles Oliveira	Samsung A34	1500	Cartão
Charles Oliveira	Notebook Gamer Lenovo LOQ 15	3500	Cartão
Virna Jandiroba	Câmera Nikon D60	1200	Cartão

NoSQL:

```
[75] # 2: Consultar quais pessoas compraram produtos acima de mil reais e como foi pago

query = pedidos.find({})

resultados = []

for pedido in query:
    cliente = clientes.find_one({"_id": pedido['cliente']['_id']})
    for produto_pedido in pedido['produtos']:
        produto = produtos.find_one({"_id": produto_pedido['_id']})
        if produto['preco'] > 1000.00:
            resultado = {
                "Nome_do_cliente": cliente['nome'],
                "Nome_do_produto": produto['nome'],
                "Preco_do_produto": produto['preco'],
                "Forma_de_pagamento": pedido['forma_pgto']
            }
            resultados.append(resultado)

for resultado in resultados:
    print(resultado)

# Explicacao: de inicio colocamos na 'query' os dados de 'pedidos', entao comecemos a trabalhar com esses dados, percorrendo o 'query' em busca de um cliente em que seu 'id' esteja em um pedido e, nesse pedido conthenha um produto...
# que tenha um preco maior que 1000 reais, entao colocamos os dados selecionados na variavel 'resultado', que eh jogada para dentro de 'resultados', que sera percorrido, imprimindo os dados selecionados.

{'Nome_do_cliente': 'Charles Oliveira', 'Nome_do_produto': 'Samsung A34', 'Preco_do_produto': 1500.0, 'Forma_de_pagamento': 'Cartão'}
{'Nome_do_cliente': 'Charles Oliveira', 'Nome_do_produto': 'Notebook Gamer Lenovo LOQ 15', 'Preco_do_produto': 3500.0, 'Forma_de_pagamento': 'Cartão'}
{'Nome_do_cliente': 'Virna Jandiroba', 'Nome_do_produto': 'Câmera Nikon D60', 'Preco_do_produto': 1200.0, 'Forma_de_pagamento': 'Cartão'}
```

3: Consultar a quantidade total de produtos de cada categoria foram comprados e o valor total

SQL:

```
SELECT
    ca.nome AS Nome_da_categoria,
    SUM(pp.quantidade) AS Quantidade_de_itens,
    SUM(pr.preco * pp.quantidade) AS Preco_total
FROM Pedido_has_produto pp
INNER JOIN Produto pr ON pp.id_produto = pr.id_produto
INNER JOIN Categoria ca ON pr.id_categoria = ca.id_categoria
GROUP BY ca.nome;
```

NOME_DA_CATEGORIA	QUANTIDADE_DE_ITENS	PRECO_TOTAL
Roupas	10	900
Eletrônicos	4	7000
Livros	7	500

NoSQL:

```
[76] # 3: Consultar a quantidade total de produtos de cada categoria foram comprados e o valor total

query = pedidos.find({})

resultados = {}

for pedido in query:
    for item in pedido['produtos']:
        produto = produtos.find_one({'_id': item['_id']})
        if produto:
            categoria_nome = produto['categoria']['nome']

            if categoria_nome not in resultados:
                resultados[categoria_nome] = {'Quantidade_de_itens': 0, 'Preco_total': 0.0}

            resultados[categoria_nome]['Quantidade_de_itens'] += item['quantidade']
            resultados[categoria_nome]['Preco_total'] += produto['preco'] * item['quantidade']

resultado_final = [
    {'Nome_da_categoria': categoria,
     'Quantidade_de_itens': data['Quantidade_de_itens'],
     'Preco_total': data['Preco_total']}
    for categoria, data in resultados.items()
]

for item in resultado_final:
    print(item)

# Explicacao: de início colocamos na 'query' os dados de 'pedidos', como na anterior, então começamos a percorrer a query em busca de produtos que fazem parte de algum pedido e ainda não foram inseridos no 'resultados' assim...
# colocando-os na variável e armazenando a quantidade e o preço, para serem usados na variável 'resultado_final', que guardará esses dados e os imprimirá no último loop.

{'Nome_da_categoria': 'Eletrônicos', 'Quantidade_de_itens': 4, 'Preco_total': 7000.0}
{'Nome_da_categoria': 'Livros', 'Quantidade_de_itens': 7, 'Preco_total': 500.0}
{'Nome_da_categoria': 'Roupas', 'Quantidade_de_itens': 10, 'Preco_total': 900.0}
```

4: Consultar os produtos com uma nota média acima de 4.0 e os clientes que os compraram

SQL:

```
SELECT
    c.nome AS Nome_do_cliente,
    pr.nome AS Nome_do_produto,
    AVG(a.nota) AS Media_de_avaliacao
FROM Cliente c
INNER JOIN Pedido p ON c.id_cliente = p.id_cliente
INNER JOIN Pedido_has_produto pp ON p.id_pedido = pp.id_pedido
INNER JOIN Produto pr ON pp.id_produto = pr.id_produto
INNER JOIN Avaliacao a ON pr.id_produto = a.id_produto
WHERE pr.id_produto IN (
    SELECT id_produto
    FROM Avaliacao
    GROUP BY id_produto
    HAVING AVG(nota) > 4.0)
GROUP BY c.nome, pr.nome;
```

NOME_DO_CLIENTE	NOME_DO_PRODUTO	MEDIA_DE_AVALIACAO
Caio Borralho	Diário de um Banana: Dias de Cão	5
Charles Oliveira	Samsung A34	4.5
Alex Pereira	Calça Jeans Boca de Sino	5
Amanda Lemos	Diário de um Banana: Dias de Cão	5
Charles Oliveira	Notebook Gamer Lenovo LOQ 15	4.8

```
[77] # 4: Consultar os produtos com uma nota média acima de 4.0 e os clientes que os compraram

produtos = db.produtos.find()

produto_media_avaliacao = {}
for produto in produtos:
    avaliacoes = produto.get('avaliacoes', [])
    if avaliacoes:
        media_avaliacao = sum(avaliacao['nota'] for avaliacao in avaliacoes) / len(avaliacoes)
        if media_avaliacao > 4.0:
            produto_media_avaliacao[produto['_id']] = {
                'nome': produto['nome'],
                'media_avaliacao': media_avaliacao
            }

resultado_final = []

for produto_id in produto_media_avaliacao:
    pedidos = db.pedidos.find({'produtos_id': produto_id})
    for pedido in pedidos:
        cliente = pedido['cliente']
        resultado_final.append({
            'Nome_do_cliente': cliente['nome'],
            'Nome_do_produto': produto_media_avaliacao[produto_id]['nome'],
            'Media_de_avaliacao': produto_media_avaliacao[produto_id]['media_avaliacao']
        })

for resultado in resultado_final:
    print(resultado)

# Explicacao: essa foi a mais complexa, de inicio colocamos em 'produtos' os valores da tabela, entao usamos um loop inicial que fara a media de notas dos pedidos que tiveram alguma avaliacao...
# com essa informacao, partimos para outro loop, que prepara os resultados relacionando os dados das medias de avaliacao com os pedidos que contem esses produtos, entao colocando ali os clientes...
# que fizeram esses pedidos.
```

```
{
  'Nome_do_cliente': 'Charles Oliveira', 'Nome_do_produto': 'Samsung A34', 'Media_de_avaliacao': 4.5
},
{
  'Nome_do_cliente': 'Charles Oliveira', 'Nome_do_produto': 'Notebook Gamer Lenovo LOQ 15', 'Media_de_avaliacao': 4.8
},
{
  'Nome_do_cliente': 'Caio Borralho', 'Nome_do_produto': 'Diário de um Banana: Dias de Cão', 'Media_de_avaliacao': 5.0
},
{
  'Nome_do_cliente': 'Amanda Lemos', 'Nome_do_produto': 'Diário de um Banana: Dias de Cão', 'Media_de_avaliacao': 5.0
},
{
  'Nome_do_cliente': 'Alex Pereira', 'Nome_do_produto': 'Calça Jeans Boca de Sino', 'Media_de_avaliacao': 5.0
}
```

5: Consultar os emails dos clientes e transportadoras cadastrados

SQL:

```
SELECT
    c.nome AS Nome,
    c.email AS Email
FROM Cliente c
UNION
SELECT
    t.nome AS Nome,
    t.email AS Email
FROM Transportadora t;
```

NOME	EMAIL
Alex Pereira	alex.pereira@ig.com
Amanda Lemos	amanda.lemos@outlook.com
Amanda Nunes	amanda.nunes@outlook.com
Caio Borralho	caio.borralho@protonmail.com
Charles Oliveira	charles.oliveira@gmail.com
Correios	contato@correios.com.br
Diego Lopes	diego.lopes@protonmail.com
Paulo Costa	paulo.costa@gmail.com
Rafael dos Anjos	rafael.anjos@ig.com
Sedex	contato@sedex.com.br
Taila Santos	taila.santos@yahoo.com
Virna Jandiroba	virna.jandiroba@yahoo.com

NoSQL:

```
[78] # 5: Consultar os emails dos clientes e transportadoras cadastrados (particularmente achei uma consulta bem sem fraca, mas acabaram as ideias)

clientes_res = list(db.clientes.find({}, {'_id': 0, 'nome': 1, 'email': 1}))

transportadoras_res = list(db.transportadoras.find({}, {'_id': 0, 'nome': 1, 'email': 1}))

resultados = clientes_res + transportadoras_res

for resultado in resultados:
    print(resultado)

# Explicacao: de inicio buscamos diretamente nos clientes e transportadoras os seus emails, entao os juntamos como uma soma, bem simples
```

```
{ 'nome': 'Charles Oliveira', 'email': 'charles.oliveira@gmail.com' }
{ 'nome': 'Amanda Nunes', 'email': 'amanda.nunes@outlook.com' }
{ 'nome': 'Alex Pereira', 'email': 'alex.pereira@ig.com' }
{ 'nome': 'Virna Jandiroba', 'email': 'virna.jandiroba@yahoo.com' }
{ 'nome': 'Caio Borralho', 'email': 'caio.borralho@protonmail.com' }
{ 'nome': 'Paulo Costa', 'email': 'paulo.costa@gmail.com' }
{ 'nome': 'Amanda Lemos', 'email': 'amanda.lemos@outlook.com' }
{ 'nome': 'Rafael dos Anjos', 'email': 'rafael.anjos@ig.com' }
{ 'nome': 'Taila Santos', 'email': 'taila.santos@yahoo.com' }
{ 'nome': 'Diego Lopes', 'email': 'diego.lopes@protonmail.com' }
{ 'nome': 'Correios', 'email': 'contato@correios.com.br' }
{ 'nome': 'Sedex', 'email': 'contato@sedex.com.br' }
```



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação e Educação Continuada
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br