

OTIMIZANDO BANCO DE DADOS ORACLE NA NUVEM ATRAVÉS DE ÍNDICES

Gabriela Grahl de Oliveira
Graduando em Banco de Dados pela Fatec Bauru
gabriela.oliveira64@fatec.sp.gov.br

Matheus Borges Ribeiro
Graduando em Banco de Dados pela Fatec Bauru
matheus.ribeiro34@fatec.sp.gov.br

Matheus Henrique Sampaio Carvalho
Graduando em Banco de Dados pela Fatec Bauru
matheus.carvalho32@fatec.sp.gov.br

Orientador: Gustavo Cesar Bruschi
Docente na Fatec Bauru
gustavo.bruschi@fatec.sp.gov.br

RESUMO

Diante do notável crescimento exponencial no volume de dados armazenados por Bancos de Dados, surge a necessidade de aprimorar não apenas a capacidade de armazenamento e manipulação dos mesmos, mas também, em específico, a eficiência na consulta pelas informações desejadas. Comumente reconhecida como uma prática essencial, a implementação de índices destaca-se como uma das técnicas mais eficazes para otimizar o desempenho de Bancos de Dados, promovendo agilidade no processamento de consultas nas tabelas. Este trabalho tem como propósito analisar os planos de execução empregados pelo banco em consultas realizadas com e sem a utilização de índices do tipo *Árvore B*Tree*, visando melhorar o desempenho e reduzir o tempo nas buscas pelas informações em um Banco de Dados em Nuvem Oracle.

Palavras-chave: banco de dados na nuvem; índice de banco de dados; otimização do banco de dados; oracle 22C.

1 INTRODUÇÃO

Com o avanço das tecnologias digitais e a expansão do consumo diário por parte de seus usuários, a produção de dados cresce de forma vertiginosa. Segundo os dados da Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos domicílios brasileiros (TIC Domicílios), a situação foi intensificada com a pandemia de Covid-19 ao apontar que a proporção de domicílios com acesso à Internet chegou a 83% em 2020, representando um aumento de 12% em relação a 2019 (Migon, *et al.*, 2021, p. 27). Tal acréscimo significativo se deveu à necessidade da continuidade das atividades empresariais com o home office, do comércio com as vendas online, da prestação de serviços, das atividades educacionais com o ensino remoto e do aumento na busca por opções de entretenimento (Nitahara, 2021). Um relatório ainda mais recente da Statista, empresa alemã especializada em coleta e visualização de dados, prevê que a quantidade de dados possa ultrapassar os 180 zetabytes até 2025 (Taylor, 2023).

É por meio deste cenário que termos como Big Data, por exemplo, vêm ganhando atenção. De acordo com o analista Edd Wilder-James (2012), ele é utilizado para descrever justamente estes extensos volumes de dados nos mais variados formatos e que são manipulados em alta velocidade. E para conseguirem lidar com este mar de informações, as organizações buscam por soluções de armazenamento, gerenciamento e transmissão com propriedades resistentes. Uma destas soluções é a escolha pelos DBaaS, do inglês *Database as a Service* (Banco de Dados como Serviço), que são os Sistemas de Gerenciamento de Banco de Dados em nuvem. Para Elmasri e Navathe (2011), é a alternativa mais eficiente aos SGBD tradicionais, que possuem limitação na utilização dos recursos e na possibilidade de escalonar, consequências da dificuldade de particionar os dados de forma eficiente. Gigantes da tecnologia como a Oracle já oferecem produtos nesse segmento, sendo um deles, o Oracle 22C.

Paralelamente, tem-se uma das técnicas mais poderosas de otimização de banco de dados sendo adotada. A criação de índices, que é aplicada para incrementar a performance ao agilizar o processamento de consultas nos registros das tabelas. A própria Oracle Database oferece diferentes tipos de índices que podem melhorar o desempenho do código SQL. No entanto, percebe-se que quando mal utilizados, diversos problemas são acarretados, como sobrecarga nas operações de inclusão, alteração e exclusão de dados. Levando em consideração a quantidade de dados armazenados, é atraente e essencial para as empresas que o resultado da busca pela informação desejada dentro de seu banco de dados venha da forma mais rápida possível.

Desta forma, tem-se a seguinte pergunta de pesquisa: **“Como alcançar o potencial máximo dos índices para otimizar um Banco de Dados em Nuvem Oracle?”**. Este trabalho tem como objetivo geral realizar um estudo comparativo entre o desempenho dos diferentes tipos de índices existentes no Oracle 22C, o Banco de Dados em Nuvem escolhido, visando atingir o potencial máximo de otimização.

2 REFERENCIAL TEÓRICO

2.1 Computação em Nuvem

A Computação em Nuvem, do inglês *Cloud Computing*, segundo a empresa Google (2023), oferece recursos configurados e escalonáveis sob demanda, permitindo aos usuários finais acessarem, com enorme facilidade, diversas aplicações e serviços quando quiserem e da onde estiverem, por meio da internet. Toda a infraestrutura e recursos computacionais ficam “ocultos”, possibilitando apenas a visualização da interface em si dessas aplicações e serviços. É este ambiente que não as pessoas não conseguem ver, capaz de processar e armazenar informações, sem depender de nenhuma máquina para existir, que remete ao uso da palavra “Nuvem” na construção do termo (Silva, 2010).

Como vantagens, tem-se, além de economia pelo pagamento apenas pelos recursos utilizados e da segurança graças aos mecanismos de segurança que os provedores de nuvem implementam e que são complexos, profundos e extensos (Google LLC, 2023), a elasticidade e o escalonamento. Ou seja, em relação a estes três últimos benefícios, os recursos são oferecidos de forma rápida, quase automática, de acordo com o aumento da demanda e retidos, no caso da diminuição da mesma (Buyya; Broberg; Goscinski, 2011).

Dentre os principais tipos de computação em nuvem, a Amazon (2023), outra empresa influente no universo da tecnologia, destaca:

- a) A Infraestrutura como serviço, do inglês *Infrastructure as a Service* (IaaS): engloba os componentes básicos da tecnologia da informação (TI) na nuvem, permitindo a utilização de recursos de rede, computadores e espaço de armazenamento de dados e proporcionando um alto grau de flexibilidade e controle de gerenciamento, visto que é o que mais se assemelha aos recursos de TI tradicionais no local. Exemplo de serviços IaaS: a própria Amazon Web Services (AWS);
- b) A Plataforma como serviço, do inglês *Platform as a Service* (PaaS): fornece todos os recursos de hardware e software necessários para o desenvolvimento de aplicativo em nuvem, melhorando a eficiência, eliminando tarefas repetitivas de gerenciamento de recursos e manutenção de software. Exemplos de serviços PaaS: Google App Engine e Oracle Cloud Platform;
- c) O Software como serviço, do inglês *Software as a Service* (SaaS): providencia aplicativos completos e gerenciados pelo próprio provedor de nuvem. Normalmente, é para o usuário final, podendo se preocupar apenas em como utilizar o software específico. Exemplos de serviços SaaS: Gmail e Dropbox.

Ademais, tem-se o Banco de Dados como Serviço, do inglês *Database as a Service* (DaaS). Um modelo que será melhor discutido no tópico seguinte, 2.2 Banco de Dados em Nuvem.

2.2 Banco de Dados em Nuvem

De acordo com a Oracle (2023), existem dois modelos principais de implementação de um banco de dados em nuvem. O primeiro deles assemelha-se a um banco de dados gerenciado internamente pela própria empresa responsável, com a diferença de que ele será

implantado na nuvem no espaço de uma máquina virtual adquirida. Já o segundo, em que se dará enfoque neste trabalho e respectivo tópico, o DBaaS, é oferecido por provedores de serviços, sendo estes os responsáveis por realizar tarefas operacionais, de manutenção e de gerenciamento do banco de dados para o usuário final.

É tido como o mais adequado para Big Data ou quando se envolve gigantescos volumes de dados em geral, visto que, de forma semelhante aos benefícios citados pela computação em nuvem como um todo, ao utilizar os Bancos de Dados como Serviço serão oferecidos maior agilidade, elasticidade, alta escalabilidade e capacidade de armazenamento, sem a necessidade de instalar, manter, atualizar, gerenciar o banco de dados ou a infraestrutura subjacente (Abourezq; Idrissi, 2016).

Para simplificar o entendimento acerca das diferenças entre os modelos acima, segue, abaixo, o Quadro 1 que contextualiza as principais diferenças entre os tipos de banco de dados em nuvens.

Quadro 1 – Modelos de Computação em Nuvem

IaaS	PaaS	DBaaS	SaaS
Aplicação	Aplicação	Aplicação	Aplicação
Dados	Dados	Dados	Dados
Tempo de execução	Tempo de execução	Tempo de execução	Tempo de execução
Middleware	Middleware	Middleware	Middleware
Sistema Operacional	Sistema Operacional	Sistema Operacional	Sistema Operacional
Virtualização	Virtualização	Virtualização	Virtualização
Servidores	Servidores	Servidores	Servidores
Armazenamento	Armazenamento	Armazenamento	Armazenamento
Rede	Rede	Rede	Rede

Organização cuida	Provedor cuida
-------------------	----------------

Fonte: Jon Bonso, 2023. Disponível em: <https://tutorialsdojo.com/azure-cloud-service-models/>. Acesso em: 03 out. 2023.

2.3 Introdução a Otimização

O termo “Otimização” pode ser entendido, conforme o dicionário de Língua Portuguesa Significados (2023), como o processo de tornar o funcionamento de um determinado produto ou serviço em “ótimo”. É possível otimizar o desempenho de uma aplicação web para farmácias, por exemplo, a fim de deixar o cadastro de seus itens muito mais rápido e a consulta muito mais simples. A lentidão nos acessos e certas operações realizadas pelo Banco de Dados pode ser atribuída a diversos fatores. Além da administração e das aplicações, o próprio banco físico, incluindo os *storages* (sistemas de armazenamento em servidores), desempenha um papel crucial. Tais elementos também afetam a otimização do banco de dados e justamente devido a este motivo que, ao selecionar um serviço de ambiente em nuvem, é essencial considerar a análise abrangente das configurações de

hardware e software do banco de dados (XP Educação, 2023).

Tratando-se especificamente da otimização de consultas no Banco de dados Oracle para melhoria de desempenho, alguns fatores são levados em consideração, como o Plano de Execução, o monitoramento e a funcionalidade do conhecido Diagnóstico de SQL, procedimento que auxilia a otimizar as instruções de SQL em aplicações na base de dados (Oracle, 2023). A documentação da Oracle ainda detalha que a mudança de memória dedicada a cada parâmetro, juntamente com a criação dos índices, são operações que são realizadas manualmente nos Sistemas de Gerenciamento de Banco de Dados (SGBD) e que com elas já é possível alcançar de maneira expressiva um desempenho superior dentro do Banco de Dados. Em outras palavras, a utilização dos recursos do serviço do *Oracle Cloud Infrastructure* (OCI) será realizada de forma eficiente para que a sua carga de trabalho atenda as demandas de desempenho, sendo dimensionada sempre que necessário. Paralelamente, é essencial que as decisões de projeto arquitetônico permitam a flexibilidade de incorporar novos serviços que possam continuar aumentando cada vez mais a eficácia do desempenho.

2.3.1 Oracle Cloud Infrastructure

Lançado em 2016, o *Oracle Cloud Infrastructure* está entre os líderes de mercado a oferecer uma gama de recursos em nuvem que incluem Análises e BI, Big Data e Data Lake, Compute (virtualização) e Serviço de banco de dados. Dentre seus clientes é possível encontrar Uber, Zoom, FedEx, Toyota e TIM Brasil. Oferece vantagens significativas para empresas e desenvolvedores ao proporcionar desempenho elevado com uma rede de baixa latência, preços competitivos e flexíveis, segurança robusta, com isolamento de rede e conformidade com diversas normas (Oracle, 2024). Ademais, é altamente escalável, facilita integrações com infraestruturas híbridas e *multicloud*, além de possuir tecnologias *open source*. Com uma infraestrutura global de data centers, garante disponibilidade, redundância e proximidade geográfica.

2.4 Plano de Execução na Base de Dados Oracle

Segundo a Oracle (2023), um Plano de Execução no Banco de dados Oracle exhibe a sequência de operações que foram executadas para realizar uma determinada consulta feita por um usuário. Ele é composto por linhas e tabelas ordenadas que são mencionadas por uma instrução. Em uma coluna é visível os métodos de acesso para cada uma destas tabelas e os métodos de combinação de linhas de maneira coerente com as relações existentes entre as diferentes colunas que são referenciadas pelo método SQL JOIN, pelas operações de dados (filtro, custo e cardinalidade que é a quantidade de registros de uma entidade que estão associados a quantos registros da outra, de cada operação), pelo particionamento das tabelas acessadas e pela instrução que utilizará uma execução paralela.

Através do Plano de Execução, fica evidente os custos de memória para realizar a operação de consulta, para analisar a informação que está sendo consultada e para retornar a mesma ao usuário. Com estes dados em mãos, o Administrador de Banco de Dados

consegue, de forma mais simples, elaborar um método, ou seja, um índice capaz de otimizar e tornar mais eficiente a busca por aquele tipo de informação.

2.5 Índices

Segundo a documentação da IBM (2021), a indexação desempenha um papel fundamental na otimização de bancos de dados, incluindo aqueles baseados na nuvem. Esta técnica melhora significativamente a velocidade de busca por dados específicos, organizando elementos essenciais de tabelas de maneira binária. Os índices são usados para aprimorar a eficácia das operações de busca, garantindo que cada entrada na tabela esteja relacionada às informações mantidas no índice, como pode ser observado na Figura 1. Ademais, nos Bancos de Dados, os índices também impõem a chave primária e outras restrições, proporcionando acesso imediato aos valores-chave das tabelas. Eles servem como referências de dados e são estratégias essenciais para otimizar o desempenho. Na prática empresarial, os analistas de negócios especificam as implementações de restrições e índices, enquanto os desenvolvedores monitoram e fazem recomendações para sua criação.

Figura 1 – Exemplo Índice

O diagrama ilustra a estrutura de um índice. À esquerda, há uma coluna rotulada 'ÍNDICE' com os valores 1, 2, 3, 4 e 5. À direita, há uma tabela com duas colunas: 'CHAVE' e 'REGISTRO'. A coluna 'CHAVE' contém os valores 1, 2, 3, 4 e 5, e a coluna 'REGISTRO' contém os nomes Matheus, Gabriela, Gustavo, Marcos e Luís. Linhas horizontais com setas apontando para a direita conectam cada número do índice à sua respectiva chave na tabela.

ÍNDICE	CHAVE	REGISTRO
1	1	Matheus
2	2	Gabriela
3	3	Gustavo
4	4	Marcos
5	5	Luís

Fonte: Elaborado pelos autores (2023).

Os índices desempenham um papel crítico no desempenho das consultas em bancos de dados SQL. Quando uma consulta inclui uma cláusula “*WHERE*” que serve como filtro para uma característica específica dos dados, a ausência de índices obriga o banco de dados a realizar uma varredura completa da tabela indicada. Esse tipo de varredura implica na leitura de cada linha até encontrar as informações especificados na cláusula do usuário. Se a tabela for extensa, conseqüentemente essa operação pode se prolongar por horas, tornando o processo muito mais lento (Oracle, 2023).

Para tabelas pequenas ou cláusulas “*WHERE*” que abrangem uma grande parte dos dados da tabela, a varredura completa pode, de fato, ser mais eficiente em termos de desempenho. A decisão de utilizar um índice ou não geralmente é tomada pelo banco de dados com base nas informações coletadas sobre as tabelas e suas linhas. Em geral, a confiabilidade da Oracle nessa tomada de decisão é a opção recomendada e mais confiável.

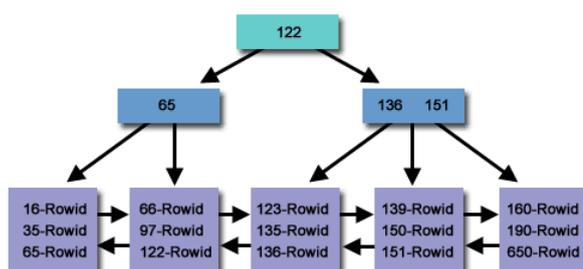
2.5.1 Tipos de Índices

Dentre as várias estruturas de índices, no que tange ao desempenho em consultas SQL, os mais utilizados são os índices do tipo Bitmap e Árvore B (B* Tree). Na hora de criar o índice, o administrador do banco de dados deve analisar a estrutura da tabela na qual as consultas serão realizadas para conseguir decidir entre um ou outro. Quantos valores diferentes compõem a tabela e qual a frequência com que os dados são atualizados simultaneamente pelos sistemas estão dentre os pontos a serem analisados nesta tomada de decisão. Usar um índice do tipo Bitmap que está associado a uma coluna inteiramente de valores únicos, por exemplo, é um recurso extremamente eficaz, assim como utilizar um índice do tipo árvore B (Sharma, 2014).

2.5.2 Índices B*Tree

O índice B*Tree utiliza uma estrutura em árvore, do termo em inglês “tree”, que separa os dados em níveis, permitindo um acesso mais eficiente. Conforme a representação da Figura 2, no topo, tem-se o nó raiz, que referencia o segundo nível, na qual têm-se vários outros nós que apontam que referenciam o terceiro nível e assim por diante. O tamanho dessa estrutura de árvore depende do número de linhas na tabela e do comprimento dos valores dos índices (Watson, 2010).

Figura 2 – Exemplo de Índice B*Tree



Fonte: Dip Mazunder, 2023. Disponível em: <https://dip-mazumder.medium.com/the-power-of-database-indexing-algorithms-b-tree-vs-hash-indexing-6e3a4112a81>. Acesso em: 05 out. 2023.

Ao realizar consultas por índices, o Oracle verifica os nós-folhas da árvore de índices, onde armazenam as chaves das linhas como ponteiros para os locais físicos correspondentes. O Oracle, assim, navega pela árvore para localizar a primeira chave correspondente e, em seguida, pelos nós-folhas do índice para encontrar outras chaves correspondentes, permitindo a recuperação ordenada das linhas das tabelas.

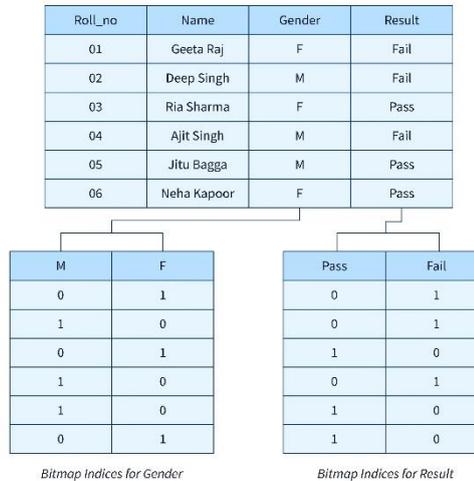
Aconselha-se usar o índice do tipo B*Tree quando a cardinalidade na coluna for alta, o número de linhas na tabela for elevado e a coluna for usada em cláusulas "WHERE" ou em condições de "JOIN".

2.5.3 Índices Bitmap

No índice do tipo Bitmap, o Oracle cria um mapa de bits no qual atribui "1" a cada linha que contenha um valor específico e "0" às linhas ausentes de valor. Esse mapa de bits abrange todas as linhas da tabela, incluindo todos os possíveis valores para a coluna

(Watson, 2010). A seguir, representado pela Figura 3, um exemplo da estrutura do índice Bitmap.

Figura 3 – Exemplo Índice Bitmap



Fonte: Scaler, 2023, <https://www.scaler.com/topics/sql/bitmap-index/>. Acesso em: 28 set. 2023.

Para aproveitar o índice Bitmap, é importante considerar se as colunas nas tabelas têm baixa cardinalidade, ou seja, se possuem poucos valores distintos. Além disso, é aconselhável que haja um alto número de linhas na tabela e que a coluna seja utilizada em operações de álgebra booleana.

3 MATERIAIS E MÉTODOS

Para a realização dos testes, era necessário escolher um provedor de serviço de banco de dados em nuvem. Optamos pelo Oracle Cloud Infrastructure (OCI) não apenas por sua reconhecida liderança de mercado, mas também por seu robusto conjunto de ferramentas de monitoramento e diagnóstico, que atuam como uma lente de aumento para nossos dados. A integração com o banco de dados Oracle foi realizada de maneira extremamente simples e eficiente. De forma rápida, conseguimos conectar, criar as tabelas necessárias, inserir os dados e realizar as consultas para as análises.

Levando em consideração a vantagem proporcionada pelos serviços em nuvem de dar ao usuário final a possibilidade de acessar a aplicação de onde estiver, os testes foram realizados em dois computadores localizados em residências diferentes e intitulados de “Máquina M” e “Máquina G” a fim de observar se isto pode ser de alguma forma um ponto de atenção para o desempenho dos índices no Banco de Dados em Nuvem. A primeira máquina contém um processador Intel Core i5-1125G de velocidade base 2,42 *gigahertz* e 4 núcleos, 8 *gigabytes* de *Random Access Memory* (RAM) e 239 *gigabytes* de *Solid State Drive* (SSD). Já a segunda máquina contém um processador AMD Ryzen 5 1600 de velocidade base 3,20 *gigahertz* e 6 núcleos, 8 *gigabytes* de *Random Access Memory* (RAM) e 932 *gigabytes* de *Solid State Drive* (SSD).

Para aproximar o cenário a um caso real, optou-se por um contexto em que o Banco de Dados Nuvem pertence a um E-commerce e desta forma foi criada uma tabela nomeada

de “temp_cli_venda” com dados fictícios de vendas e clientes utilizando o Oracle SQL Developer, sendo a sua estrutura a descrita na (Figura 4).

Figura 4 - Criação da Tabela

```
CREATE TABLE TEMP_CLI_VENDA (  
    VENDA_COD NUMBER(10),  
    PRODUTO_COD NUMBER(10),  
    DATA_VENDA DATE,  
    VALOR_VOTAL NUMBER(10,2),  
    CLI_NOME VARCHAR2(50),  
    CLI_SALDO NUMBER(6,2),  
    CLI_EMAIL VARCHAR2(100),  
    CLI_TELEFONE VARCHAR2(15),  
    CLI_ENDERECO VARCHAR2(100),  
    CLI_CIDADE VARCHAR2(50),  
    CLI_ESTADO VARCHAR2(50)  
);
```

Fonte: Elaborado pelos autores, 2024

Ela foi povoada com quinhentos mil registros aleatórios gerados através de Procedural Language/Structured Query Language (PL/SQL) (Figura 5).

Figura 5 - Inserindo Dados na Tabela

```
BEGIN  
FOR i IN 1..500000 LOOP  
    INSERT INTO TEMP_CLI_VENDA VALUES (  
        i,  
        ROUND(dbms_random.value(1,100)),  
        SYSDATE - ROUND(DBMS_RANDOM.VALUE(1, 365)),  
        ROUND(DBMS_RANDOM.VALUE(10, 1000), 2),  
        dbms_random.string('U', 50),  
        dbms_random.value(0, 1000),  
        dbms_random.string('U', 100),  
        dbms_random.string('N', 15),  
        dbms_random.string('U', 100),  
        dbms_random.string('U', 50),  
        dbms_random.string('U', 50)  
    );  
    IF MOD(i, 100) = 0 THEN  
        COMMIT;  
    END IF;  
END LOOP;  
COMMIT;  
END;
```

Fonte: Elaborado pelos autores, 2024

3.2 Métricas Utilizadas

A análise das consultas foi baseada no Plano de Execução que é gerado pela própria ferramenta Oracle SQL Developer assim que uma consulta é executada e foi empregado cinco métricas para avaliar e otimizar o desempenho do banco de dados. Essas métricas incluem:

- Time (Tempo de Acesso):** Refere-se ao tempo total necessário para executar o Plano de Execução. Isso inclui o tempo de I/O (do inglês, *Input/Output*) o tempo de CPU e qualquer outro tempo de espera ou processamento associado à execução da consulta.
- CPU Cost (Custo da CPU):** Indica o custo computacional necessário para executar operações no plano de execução. Isso inclui operações de filtragem, junção, ordenação e qualquer outra operação que exija processamento na CPU.
- I/O Cost (Custo de Entrada/Saída):** Refere-se ao custo associado à leitura e gravação de

dados no disco. Quanto maior o número de operações de entrada e saída, maior será o custo em termos de tempo e recursos.

- d) Cost: É uma métrica relativa que leva em consideração outros fatores como o próprio uso da CPU e operações de I/O citadas anteriormente. Planos que possuem menor custo são geralmente preferidos pelo otimizador de consultas por serem considerados mais eficientes em termos de uso de recursos.

Reduzir qualquer uma das métricas citadas anteriormente pode resultar em consultas mais eficazes, otimizando o desempenho do Banco de Dados.

3.3 Condução dos Testes

Para conduzir os testes, foi empregado dez tipos distintos de consultas, utilizando colunas e cláusulas de filtragem diferentes. É importante notar que cada uma dessas consultas teve uma versão com índice e outra sem. Foi escolhido os índices do tipo B*Tree, padrão do Oracle, devido à sua alta eficiência em aplicações OLTP, do inglês tablespaces, nas quais as consultas dos usuários tendem a ser relativamente repetitivas. Os índices criados no formato B*Tree tendem a serem mais performáticos quando os dados estão bem depurados antes da implantação do ambiente de produção. Para bases de dados com constante inserção de dados, essa característica dos índices de B*Tree se destaca como a escolha mais adequada, visando performance e otimização

4 RESULTADOS OBTIDOS

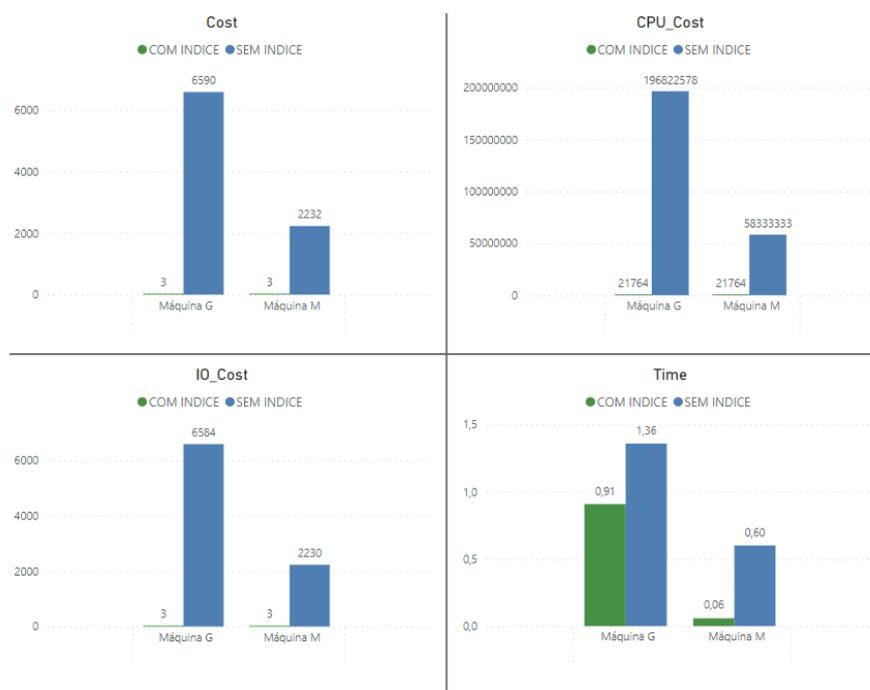
A Consulta A envolveu a coluna cli_estado que contém uma alta cardinalidade. Foi filtrado por um estado específico, tendo um retorno de sete linhas no total. A utilização do índice conseguiu a maior economia média de todas as consultas, chegando a quase 100% para todos os custos, levando em consideração ambas as máquinas utilizadas, como observado nos resultados registrados na Tabela 1. Em relação ao tempo, foi possível obter uma economia média de 30%. Questão de milissegundos.

Tabela 1 – Média dos Resultados da Consulta A

CONSULTA A		Cost	CPU_Cost	IO_Cost	Time
SELECT VENDA_ID, CLI_ESTADO FROM TEMP_CLI_VENDA WHERE CLI_ESTADO = 'SP'	SEM INDICE	4411	127577955,5	4407	0,6795
	COM INDICE	3	21764	3	0,454

Fonte: Elaborado pelos autores, 2024

Gráfico 1 – Gráfico das Métricas da Consulta A



Fonte: Elaborado pelos autores, 2024

A Consulta B envolveu a coluna valor_total que contém uma cardinalidade média. Foi filtrado por valores que estivessem acima de quinhentos, tendo um retorno de duzentos e cinquenta e dois mil duzentos e seis linhas no total. Neste caso, o uso do índice forçou a utilização da CPU. Em contrapartida, a economia do custo geral e do custo de I/O foram consideravelmente relevantes, atingindo a média de 73% em cada um. Já o tempo, foi interessante notar que para a máquina M, a diferença, apesar de pequena, foi positiva, enquanto para a máquina G foi negativa.

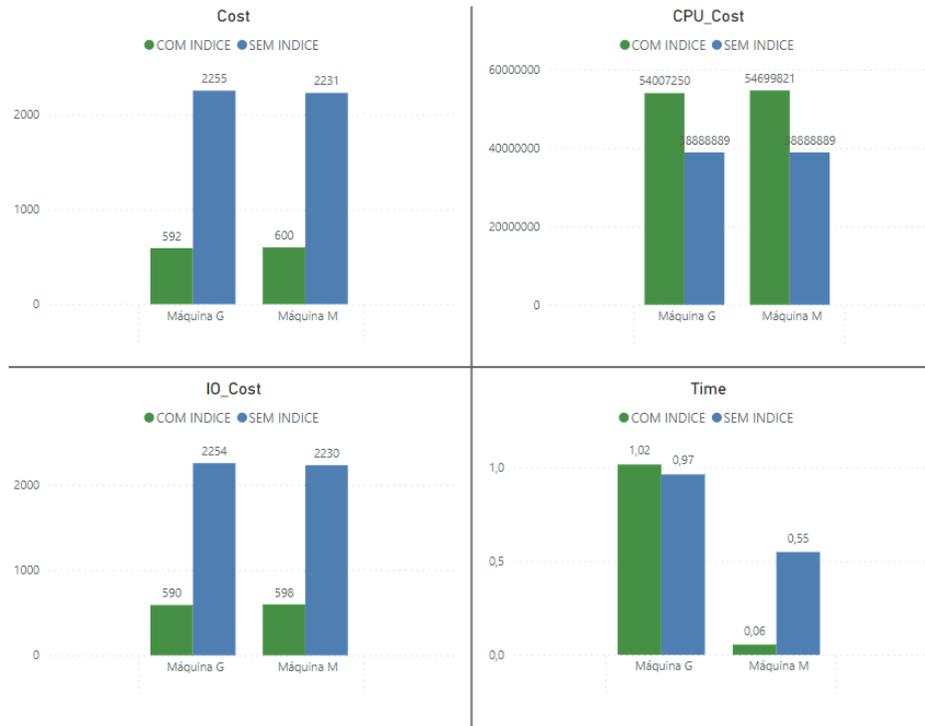
A Tabela 2 apresenta em detalhes os resultados obtidos.

Tabela 2 – Média dos Resultados da Consulta B

CONSULTA B		Cost	CPU_Cost	IO_Cost	Time
SELECT VENDA_ID, VALOR_TOTAL FROM TEMP_CL_VENDA WHERE VALOR_TOTAL > 500	SEM INDICE	2243	38888889	2242	0,7575
	COM INDICE	596	54353535,5	594	0,5365

Fonte: Elaborado pelos autores, 2024

Gráfico 2 – Gráfico das Métricas da Consulta B



Fonte: Elaborado pelos autores, 2024

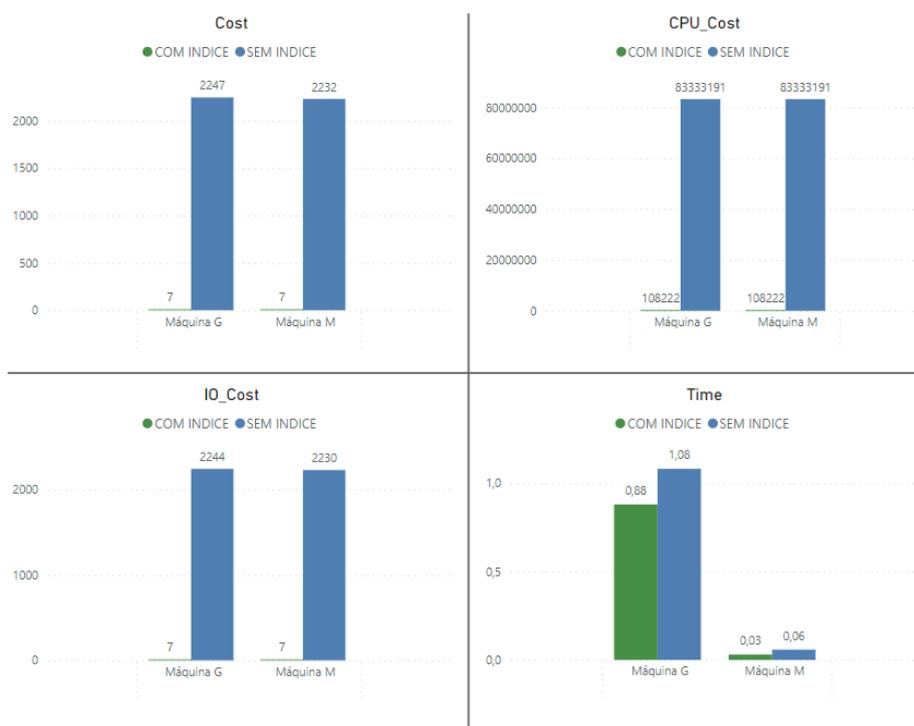
A Consulta C envolveu a coluna cli_cidade que contém uma alta cardinalidade. Foi filtrado por cinco cidades específicas, tendo um retorno de trinta e uma linhas no total. Na query, foi optado pelo operador *IN* que também tende a afetar a performance da consulta. Ele verifica se o valor está contido em uma lista de valores. Novamente, foi possível obter uma economia média de 99,90% para todos os custos, levando em consideração ambas as máquinas. A máquina M apresentou uma melhor diferença de tempo, porém, também foi possível notar que a pesquisa foi mais rápida para a máquina G, como visualizado nos resultados registrados na Tabela 3.

Tabela 3 – Média dos Resultados da Consulta C

CONSULTA C		Cost	CPU_Cost	IO_Cost	Time
SELECT * FROM TEMP_CLI_VENDA_GABI WHERE CLI_CIDADE IN ('Bauru', 'Botucatu', 'Águdos', 'Piratininga', 'Jaú')	SEM INDICE	2239,5	83333191	2237	0,57
	COM INDICE	7	108222	7	0,455

Fonte: Elaborado pelos autores, 2024

Gráfico 3 – Gráfico das Métricas da Consulta C



Fonte: Elaborado pelos autores, 2024

A Consulta D envolveu a coluna cli_cidade e a coluna cli_estado. Ambas contêm uma alta cardinalidade. Foi filtrado por uma cidade de um estado específico, usando a cláusula *AND* na filtragem. Foi retornado seis linhas no total. Neste quarto experimento, optou-se por colocar dois índices, um em cada coluna. De forma geral, a economia também foi positiva, alcançando o patamar da média de 99,90% para todas as métricas de custo e um recorde de economia de tempo para a máquina M, que chegou a 88%, apontado na Tabela 4.

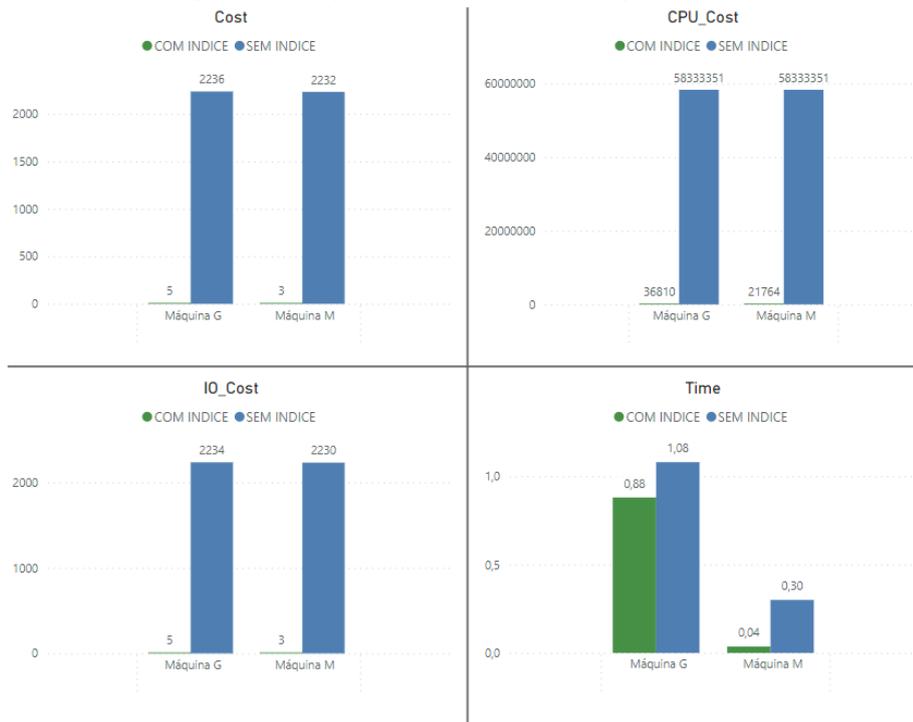
Vale ressaltar que mesmo com dois índices na tabela, o Banco optava apenas por um, sendo aquele que apresentava melhor desempenho.

Tabela 4 – Média dos Resultados da Consulta D

CONSULTA D		Cost	CPU_Cost	IO_Cost	Time
<pre>SELECT *FROM TEMP_CLI_VENDA_GABI WHERE CLI_CIDADE = 'Bauru' AND CLI_ESTADO = 'São Paulo'</pre>	SEM INDICE	2234	58333351	2232	0,69
	COM INDICE	4	29287	4	0,458

Fonte: Elaborado pelos autores, 2024

Gráfico 4 – Gráfico das Métricas da Consulta D



Fonte: Elaborado pelos autores, 2024

A Consulta E também envolveu a coluna cli_cidade e a coluna cli_estado. Da mesma forma que a anterior, em ambos, foi construído um índice. A diferença aqui para as consultas C e D foram as cláusulas utilizadas na filtragem: a combinação do AND e do OR. Assim, foi solicitado ao banco retornar uma cidade de um estado específico ou outra cidade específica, resultando em treze linhas no total. Apesar da tentativa de deixar a consulta mais complexa, os índices também se desempenharam bem e trouxeram uma média de economia de 99,90% para todas as métricas de custo e um segundo recorde de economia de tempo para a máquina M, que chegou a 90%.

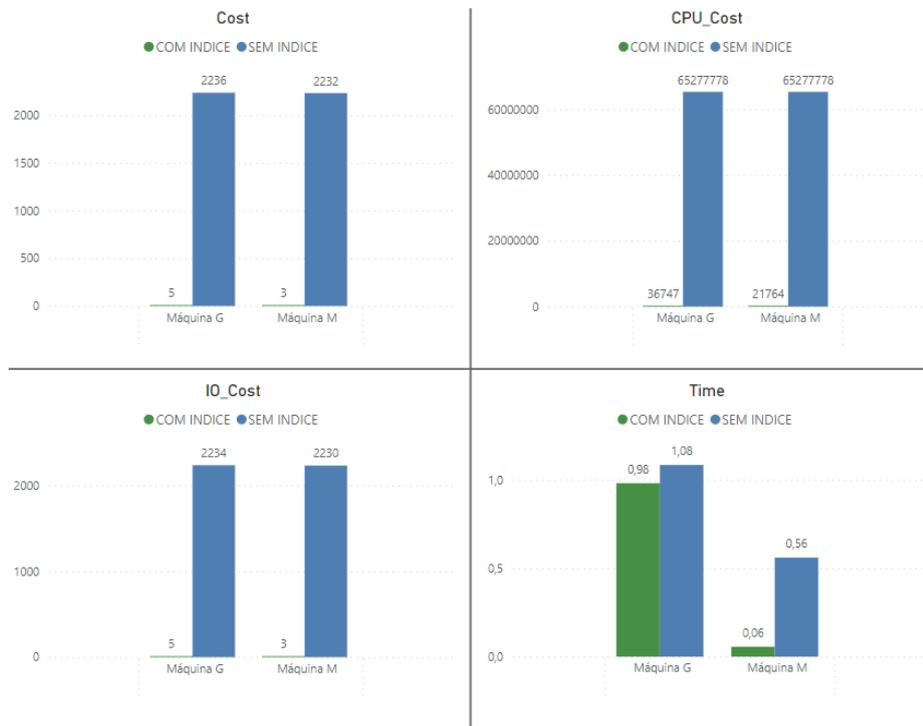
A Tabela 5 apresenta em detalhes os resultados obtidos.

Tabela 5 – Média dos Resultados da Consulta E

CONSULTA E		Cost	CPU_Cost	IO_Cost	Time
SELECT * FROM TEMP_CLI_VENDA_GABI WHERE (CLI_CIDADE = 'Bauru' AND CLI_ESTADO = 'São Paulo') OR CLI_CIDADE = 'Rio de Janeiro'	SEM INDICE	2234	65277778	2232	0,822
	COM INDICE	4	29255,5	4	0,5185

Fonte: Elaborado pelos autores, 2024

Gráfico 5 – Gráfico das Métricas da Consulta E



Fonte: Elaborado pelos autores, 2024

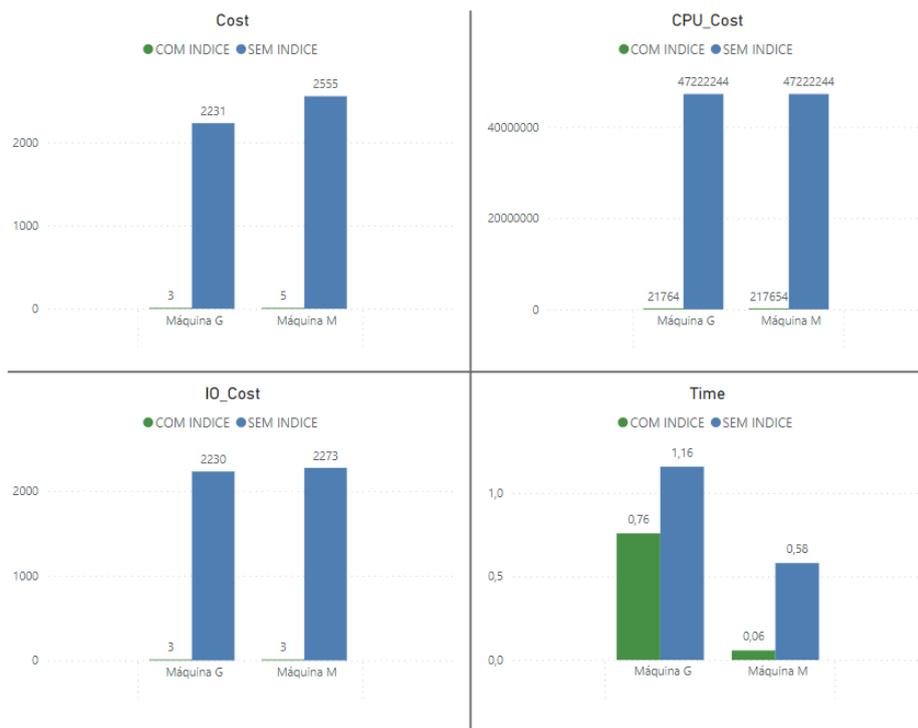
A Consulta F envolveu a coluna cli_email que contém uma cardinalidade alta. Foi filtrado por um e-mail específico, tendo um retorno de três linhas no total. A utilização do índice aqui também foi benéfica e houve uma economia também de 99,90%. Para a máquina G, houve um recorde de rapidez de tempo, como visualizado nos resultados apresentados pela Tabela 6.

Tabela 6 – Média dos Resultados da Consulta F

CONSULTA F		Cost	CPU_Cost	IO_Cost	Time
SELECT CLI_NOME, CLI_CIDADE, CLI_ESTADO FROM TEMP_CLI_VENDA WHERE CLI_EMAIL = 'contato@fatec.com.br';	SEM INDICE	2393	47222244	2251,5	0,8685
	COM INDICE	4	119709	3	0,4075

Fonte: Elaborado pelos autores, 2024

Gráfico 6 – Gráfico das Métricas da Consulta F



Fonte: Elaborado pelos autores, 2024

A Consulta G envolve a coluna cli_saldo que contém uma cardinalidade média. Foi utilizada a cláusula *BETWEEN*, filtrando por valores específicos, que deviam estar entre o número duzentos e cinquenta e o número novecentos. Houve um retorno de zero linhas. Para quebrar o padrão de resultados positivos, este foi um caso de grande divergência nas métricas obtidas e que podem ser mais bem analisadas na Tabela 6. Não houve economia de custo para a máquina G. A CPU foi mais utilizada e o custo geral e o custo de I/O foram superiores, utilizando o índice. Apenas houve uma rapidez, embora extremamente pequena, de tempo.

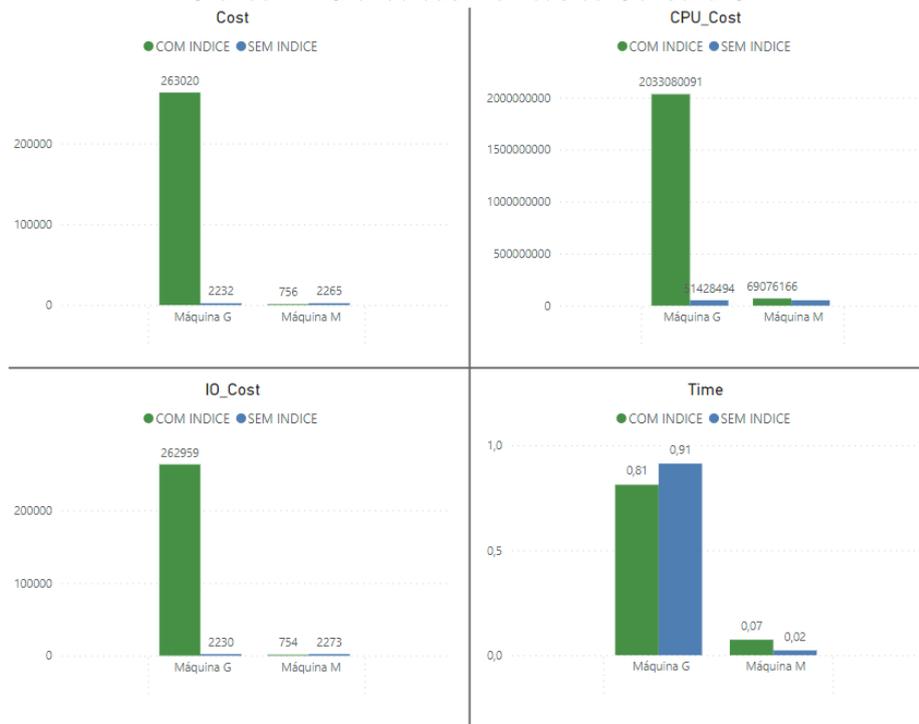
De forma semelhante, a utilização da CPU na máquina M também foi mais requisitada. Em contrapartida, até houve uma economia de custo geral e custo de I/O, mas que ficaram bem abaixo das demais consultas, tendo uma média de 66% para cada um. Para esta máquina, foi mais lento usar a consulta com índice.

Tabela 7 – Média dos Resultados da Consulta G

CONSULTA G		Cost	CPU_Cost	IO_Cost	Time
SELECT CLI_NOME, CLI_EMAIL, CLI_COD FROM TEMP_CLI_VENDA WHERE CLI_SALDO BETWEEN 250 AND 900;	SEM INDICE	2248,5	51427124,5	2251,5	0,468
	COM INDICE	131888	1051078129	131856,5	0,443

Fonte: Elaborado pelos autores, 2024

Gráfico 7 – Gráfico das Métricas da Consulta G



Fonte: Elaborado pelos autores, 2024

A Consulta H envolve a coluna valor_total e foi a que possui a query mais bem elaborada. Nela é solicitado retornar uma lista de combinações com a soma dos valores totais e a data da última venda para cada um dos agrupamentos que é feito. Ademais, a filtragem pede por linhas em que o valor total seja superior a cem.

Os resultados foram muito semelhantes à Consulta G, no quesito de trazer prejuízo de performance e custo para a máquina G ao se utilizar o índice. Houve uma maior utilização da CPU e o encarecimento do custo geral e do custo de I/O. A consulta só ficou um pouco mais rápida.

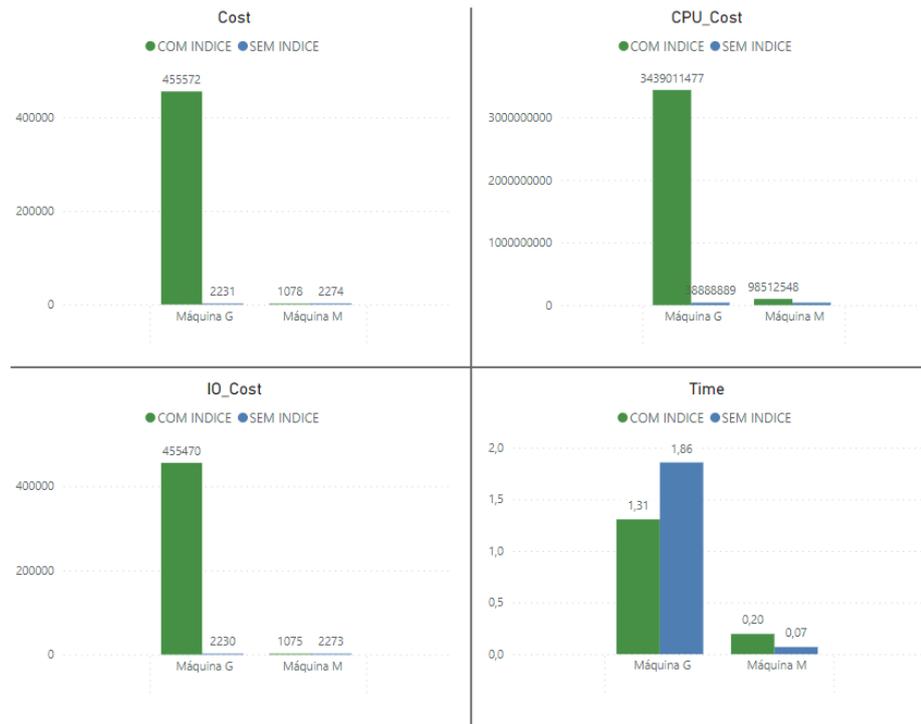
Enquanto na máquina M, o uso da CPU e do tempo para concluir a consulta também foram maiores. A economia geral e a economia de I/O foram as menores registradas de todos os testes, chegando a uma média de 52% para cada um.

Tabela 8 – Média dos Resultados da Consulta H

CONSULTA H		Cost	CPU_Cost	IO_Cost	Time
SELECT PRODUTO_ID, VENDA_ID, CLL_COD, SUM(VALOR_TOTAL) AS TOTAL, MAX(DATA_VENDA) AS DATA FROM TEMP_CLL_VENDA WHERE VALOR_TOTAL > 100 GROUP BY PRODUTO_ID, VENDA_ID, CLL_COD;	SEM INDICE	2252,5	38888889	2251,5	0,963
	COM INDICE	228325	1768762013	228272,5	0,751

Fonte: Elaborado pelos autores, 2024

Gráfico 8 – Gráfico das Métricas da Consulta H



Fonte: Elaborado pelos autores, 2024

Para as duas últimas consultas, foi decidido separar os dados da “temp_cli_venda” em duas tabelas diferentes. A primeira foi nomeada "cliente" e foi povoada com um milhão de registros aleatórios. Sua estrutura está a representada pela Figura 6.

Figura 6 - Criação da Tabela Cliente

```
CREATE TABLE CLIENTE (
  CLI_COD NUMBER(10),
  CLI_NOME VARCHAR2(50),
  CLI_LOGIN VARCHAR2(16),
  CLI_SENHA VARCHAR2(16),
  CLI_SB NUMBER(1),
  CLI_SALDO NUMBER(6,2),
  CLI_EMAIL VARCHAR2(100),
  CLI_TELEFONE VARCHAR2(15),
  CLI_ENDERECO VARCHAR2(100),
  CLI_CIDADE VARCHAR2(50),
  CLI_ESTADO VARCHAR2(50)
);
```

Fonte: Elaborado pelos autores, 2024

Já a segunda tabela, denominada “vendas”, foi povoada com quinhentos mil registros aleatórios, mantendo os mesmos códigos de clientes, possibilitando a ligação entre ambas por meio do JOIN, que é uma operação que combina registros de duas tabelas baseando-se em uma coluna comum. Sua estrutura está representada pela Figura 7.

Figura 7 - Tabela Vendas

```
CREATE TABLE VENDAS (
  VENDA_ID NUMBER(10),
  CLI_COD NUMBER(10),
  PRODUTO_ID NUMBER(10),
  DATA_VENDA DATE,
  VALOR_TOTAL NUMBER(10,2)
);
```

Fonte: Elaborado pelos autores, 2024

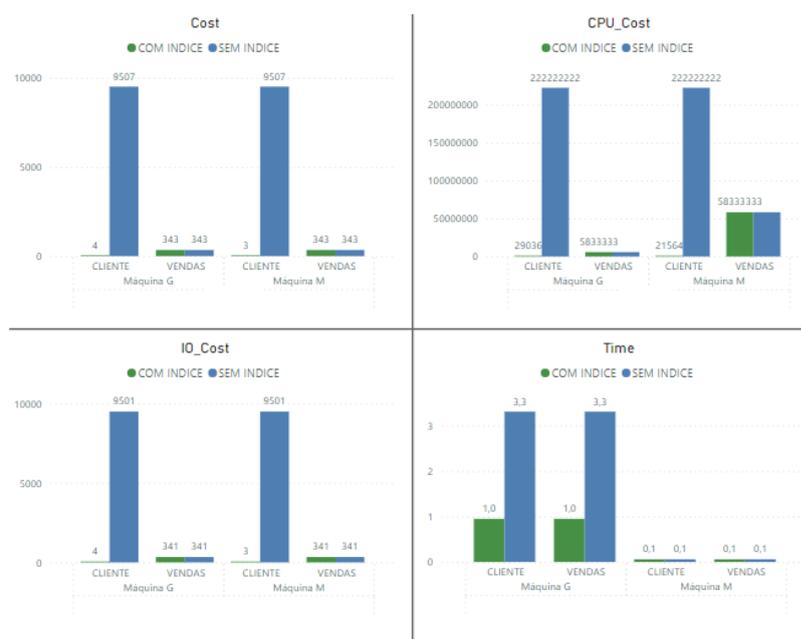
A Consulta I utilizou a coluna cli_estado da tabela “cliente”. Houve economias iguais para ambas as máquinas, chegando na média padrão de 99,90% para cada uma das métricas. Como não havia nenhum índice na tabela “vendas” e nenhum filtro utilizado para seus dados, não houve diferença de custo. A Tabela 9 apresenta em detalhes os resultados obtidos.

Tabela 9 – Média dos Resultados da Consulta I

CONSULTA I	Tabela	Cost	CPU_Cost	IO_Cost	Time	
SELECT VENDA_ID, CLI_CIDADE, CLI_ESTADO, DATA_VENDA FROM CLIENTE CLI LEFT JOIN VENDAS VEN ON VEN.CLI_COD = CLI.CLI_COD WHERE CLI_ESTADO='SP'	SEM INDICE	CLIENTE	9507	222222222	9501	1.68
	SEM INDICE	VENDAS	343	32083333	341	1.68
	COM INDICE	CLIENTE	3.5	25300	3.5	0.50
	COM INDICE	VENDAS	343	32083333	341	0.50

Fonte: Elaborado pelos autores, 2024

Gráfico 9 – Gráfico das Métricas da Consulta I



Fonte: Elaborado pelos autores, 2024

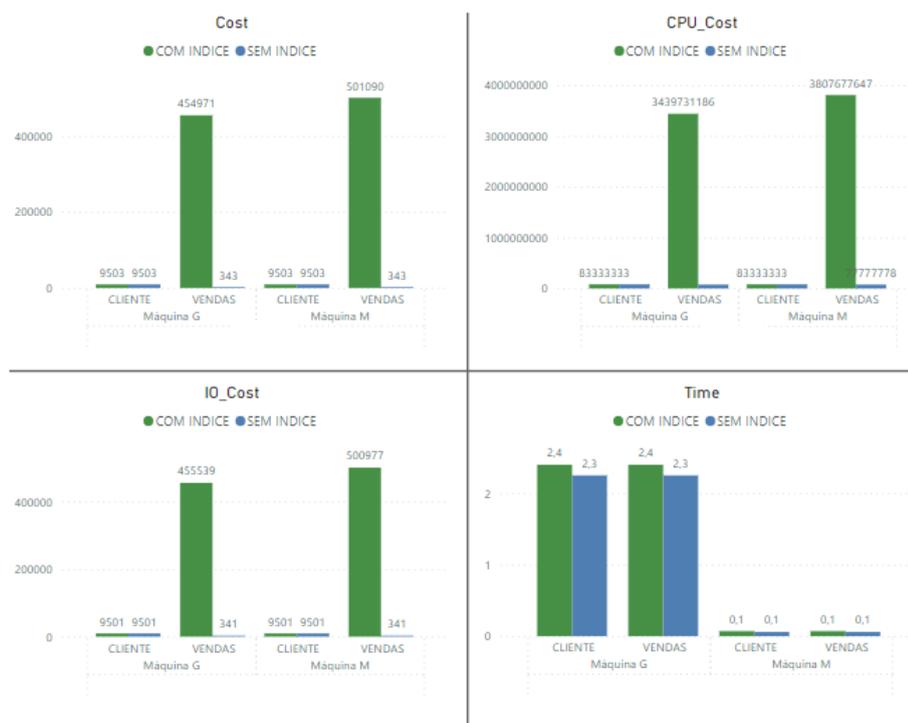
Por último, tem-se os resultados da Consulta J que envolveram a coluna valor_total da tabela “vendas”. O uso do índice prejudicou a performance da consulta e acabou acarretando prejuízo tanto de custo geral, CPU, custo de I/O e até mesmo de tempo tanto para a máquina M quanto para a máquina G como pode ser averiguado na Tabela 10 abaixo.

Tabela 10 – Média dos Resultados da Consulta J

CONSULTA J		Tabela	Cost	CPU_Cost	IO_Cost	Time
<pre> SELECT PRODUTO_ID, VENDA_ID, VEN.CLI_COD, SUM(VALOR_TOTAL)AS VALOR_TOTAL, MAX(DATA_VENDA) AS DATA FROM CLIENTE CLI LEFT JOIN VENDAS VEN ON VEN.CLI_COD = CLI.CLI_COD WHERE VALOR_TOTAL > 100 GROUP BY PRODUTO_ID, VENDA_ID, VEN.CLI_COD; </pre>	SEM INDICE	CLIENTE	9503	83333333	9501	1.15
	SEM INDICE	VENDAS	343	77777778	341	1.15
	COM INDICE	CLIENTE	9503	83333333	9501	1.24
	COM INDICE	VENDAS	478030.5	3623704417	478258	1.24

Fonte: Elaborado pelos autores, 2024

Gráfico 10 – Gráfico das Métricas da Consulta J



Fonte: Elaborado pelos autores, 2024

Finalizado as análises, averiguamos de modo gerou houve uma economia de custos positiva. Todavia, observamos que talvez seja melhor evitar colunas de valores numéricos que tendem a trazer e permitir queries que utilizam cláusulas de filtragem mais abrangentes, como OR, maior ou menor, visto que isso aumenta a cardinalidade e o número de linhas a serem trazidas torna-se ilimitado. Ambos os cenários, quando apareceram, mostraram que o índice pode perder sua eficiência e acabaram por degradar o desempenho das consultas ao invés de otimizar. Em contrapartida, optar por colunas de string, nas quais é mais provável

que a cláusula de filtragem seja mais específica, reduzindo a cardinalidade e trazendo um número de linhas mais limitado e quase previsível, pode ser uma melhor escolha. Com estas duas premissas, já é possível se direcionar com mais assertividade ao estudar quais colunas de uma tabela vale a pena investir na construção de um índice. Ademais, foi analisado que utilizar dois índices de uma mesma tabela em uma consulta não necessariamente dobra a economia ou a velocidade da pesquisa. Pelo contrário, o próprio otimizador de consulta do banco tende a escolher apenas o índice que apresentar melhor desempenho. Assim, levando em consideração que os índices também consomem espaço para armazenamento no banco e afetam a performance em escritas - como INSERT, UPDATE, DELETE - é possível escolher apenas as colunas estratégicas, cujas informações são muito utilizadas nas consultas.

Outro fator que pode condicionar os resultados é o tipo de processador. Nos testes realizados, é possível que o i5 1125G, presente na Máquina M, tenha sido mais eficiente energeticamente do que o Ryzen 5 1600, presente na Máquina G, devido ao fato de ter sido projetado especificamente para dispositivos portáteis como notebooks, além de ser mais recente do que sua concorrente. Outro detalhe técnico é a sua superioridade em desempenho por núcleo do hardware da Intel ao modelo da AMD, mesmo tendo dois núcleos a menos. Levando em consideração as particularidades das CPUs utilizadas, é evidente que cada processador lidou de forma diferente com algumas tarefas específicas divergindo nos resultados como pode ser observado no Gráficos 7 e no Gráfico 8, por exemplo. Ou seja, é válido que a empresa ou o profissional observe bem estes pontos ao utilizar os serviços do OCI, lembrando que há a opção das máquinas virtuais, nas quais a própria Oracle disponibiliza para criação e melhor adaptação da máquina de acordo com seu propósito.

5 CONSIDERAÇÕES FINAIS

Em meio a um mundo globalizado, gerador e processador de massas de informações por minuto, é natural que as empresas acelerem sua corrida contra o tempo para incrementar não apenas a estrutura, mas a performance de seus bancos de dados que, agora, também podem se encontrar na nuvem. A criação de índices é dita como uma das técnicas mais poderosas de otimização, sendo frequentemente adotada, porém, fica a dúvida se é e como é possível alcançar seu potencial máximo de otimização.

Neste estudo, foram realizados, em dois computadores, diversos testes de consultas sem e com índices, que foram construídos em diferentes colunas. Por meio dos Planos de Execução que foram gerados pelo próprio Banco de Dados em Nuvem da Oracle, foi possível analisar o seu desempenho e quais os impactos positivos e/ou negativos os índices trouxeram. Resultou que, de forma abrangente, de fato, a sua utilização, além de agilizar a busca pela informação desejada, acarreta uma economia expressiva de uso de CPU, operações I/O e recursos em geral, podendo gerar uma redução de mais de 90%. No entanto,

foi observado que a escolha pela coluna errada para sua construção pode agravar o custo. Isso vale para o tipo de processador presente na máquina a ser utilizada.

Portanto, para conseguir alcançar o potencial máximo dos índices que, de fato são poderosas ferramentas de otimização para um Banco de Dados em Nuvem, é de extrema importância estudar as colunas em que eles serão construídos, analisando fatores como o formato da informação e a maneira como ela é consumida nas consultas pelos usuários.

E para finalizar as nossas considerações finais, sugerimos a exploração dos serviços de banco de dados em nuvem oferecidos pela Amazon Web Services (AWS) em trabalhos futuros. Esta plataforma, reconhecida por sua abrangência e inovação, conta com ferramentas avançadas de monitoramento que permitem observar cada milissegundo de execução das consultas em SQL. As métricas detalhadas, gráficos analíticos e alertas de monitoramento oferecidos pela AWS são recursos valiosos. O Amazon RDS Performance Insights, por exemplo, é um dos serviços da AWS que permite identificar rapidamente problemas de desempenho, auxiliando os profissionais na busca por soluções de otimização. Entender em profundidade o funcionamento dessas ferramentas, a eficácia dos índices neste ambiente e a disponibilidade de outros métodos de otimização de consultas podem ser ideais de grande valor para complementar e aprimorar os resultados obtidos neste trabalho.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ABOUREZQ, Manar; IDRISSE, Abdellah. Database-as-a-Service for Big Data: An Overview. **International Journal of Advanced Computer Science and Applications**, Rabat, vol. 7, no. 1, 2016. Disponível em:

<https://pdfs.semanticscholar.org/a0e7/e17762d0297f9a4cb7c714aa1ee86962c512.pdf>.

Acesso em: 03 out. 2023.

AMAZON. O que é computação em nuvem?. **Amazon**, 2023. Disponível em:

<https://aws.amazon.com/pt/what-is-cloud-computing/>. Acesso em: 03 de out. de 2023.

BOTTORFF, Cassie; J.D., Belle Wong. *Top Social Media Statistics And Trends Of 2023*.

Forbes, 2023. Disponível em: <https://www.forbes.com/advisor/business/social-media-statistics/#source>. Acesso em: 28 set. 2023.

BUYA, Rajkumar; BROBERG, James; GOSCINSKI, Andrzej. **Cloud Computing – Principles and Paradigms**. 1. ed. New Jersey, U.S.: John Wiley & Sons, 2011. 664p. ISBN: 978-0-470-88799-.

ELMASRI, Ramez; NAVATHE, Shambkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson, 2011.

GOOGLE LLC. O que é a computação em nuvem?. **Google LLC**, 2023. Disponível em:

<https://cloud.google.com/learn/what-is-cloud-computing?hl=pt-br>. Acesso em: 03 out. 2023.

IBM. Indexação de Banco de Dados. **IBM**, 2021. Disponível em:

<https://www.ibm.com/docs/pt-br/control-desk/7.6.1?topic=databases-database-indexing>.

Acesso em: 28 set. 2023

MIGON, Marcio *et al.* **TIC DOMICÍLIOS 2020**. São Paulo: CGI.BR, 2021. *E-book*. Disponível em:

https://cetic.br/media/docs/publicacoes/2/20211124201233/tic_domicilios_2020_livro_eletronico.pdf. Acesso em: 28 set. 2023.

NITAHARA, Akemi. Estudo mostra que pandemia intensificou uso das tecnologias digitais.

Agência Brasil, Rio de Janeiro, 25 de nov. de 2021. Disponível em:

<https://agenciabrasil.ebc.com.br/geral/noticia/2021-11/estudo-mostra-que-pandemia-intensificou-uso-das-tecnologias-digitais>. Acesso em: 28 set. 2023

ORACLE. O que é um banco de dados em nuvem?. **Oracle**, 2023. Disponível em:

<https://www.oracle.com/br/database/what-is-a-cloud-database/>. Acesso em: 03 out. 2023.

ORACLE. Sobre a Eficiência do Desempenho e as Práticas de Otimização de Custos.

Oracle, 2023. Disponível em: <https://docs.oracle.com/pt-br/solutions/oci-best-practices/performance-efficiency-and-cost-optimization-practices.html#GUID-26B708E2-30EF-42E6-871C-63AAD0B7E6D5>. Acesso em: 06 out. 2023

ORACLE. *IT Executive's Guide to Oracle Cloud Infrastructure*. **Oracle**, 2024. Disponível em:

<https://www.oracle.com/a/ocom/docs/cloud/oracle-cloud-infrastructure-platform-overview-wp.pdf>. Acesso em: 07 jun. 2024

XP EDUCAÇÃO. Otimização de banco de dados: aprenda a elevar a performance!. **XP Educação**, 2022. Disponível em: <https://blog.xpeducacao.com.br/otimizacao-de-banco-de-dados/#:~:text=A%20otimiza%C3%A7%C3%A3o%20de%20bancos%20de,Confira%20todas%20as%20op%C3%A7%C3%B5es!&text=A%20otimiza%C3%A7%C3%A3o%20de%20banco%20de,resultados%20de%20consultas%20e%20an%C3%A1lises>. Acesso em: 06 out. 2023.

SHARMA, Vivek. Índice de bitmap x índice de árvore B: Qual e quando usar?. **Oracle**, 2014. Disponível em: <https://www.oracle.com/br/technical-resources/articles/sql/bitmap-tree-b.html>. Acesso em: 05 out. 2023.

SIGNIFICADOS. Significado de Otimização. **Significados**, 2023. Disponível em: <https://www.significados.com.br/otimizacao/>. Acesso em: 06 out. 2023

SILVA, F. H. R. **Um estudo sobre os benefícios e os riscos de segurança na utilização de Cloud Computing**; 2010. 15f. Artigo científico de conclusão de curso apresentado no Centro Universitário Augusto Motta, UNISUAM-RJ.

TAYLOR, Petroc. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025. **Statista**, 2023. Disponível em: <https://www.statista.com/statistics/871513/worldwide-data-created/>. Acesso em: 28 set. 2023.

WATSON, John. **OCA Oracle Database 11g: Administração I: Guia do Exame 1Z0-052**. 1. ed. São Paulo: Bookman, 2010. 355 p.

WILDER-JAMES, Edd. *What is big data?*. **O'Reilly**, 2012. Disponível em: <https://www.oreilly.com/ideas/what-is-big-data>. Acesso em: 28 set. 2023.