Performance de diferentes modelagens de dados em arquitetura de armazenamento moderna.

Gean Lucas Pannellini[1]*; Diego Pedroso dos Santos[2]

[1] BHub Serviços e Tecnologia LTDA. Senior Data Analyst. Rua Cardeal Arcoverde, 2365 - Terceiro andar - Pinheiros, São Paulo - SP, 05407-003, Brasil.
[2] Mestre em Engenharia de Controle e Automação. Avenida Marques de São Vicente, 2898 – Água Branca; 05034-040 São Paulo, São Paulo, Brasil
*autor correspondente: geanpanne@gmail.com

Performance de diferentes modelagens de dados em arquitetura de armazenamento moderna.

Resumo

Para lidar com um cenário de aumento significativo no uso de dados pelas empresas, novas ferramentas de armazenamento de dados surgem focando em velocidade e eficiência. Desse modo, conceitos de Inmon e Kimball sobre modelagens de dados, datados do final do século vinte e ainda significativamente aplicados nos dias atuais, se tornam pontos de questionamentos, tendo em vista que o desempenho dessas novas tecnologias podem gerar consultas resultantes de "pipeline's" de dados mais performáticas sem a específica necessidade de possuir modelagens. Portanto, o presente estudo tem como objetivo analisar o comportamento de uma consulta de dados utilizando dois cenários: o primeiro tendo como fonte das consultas tabelas normalizadas e o outro cenário consultando diretamente uma tabela desnormalizada, ambas utilizando como fonte arquivos "parquet". A base de dados foi retirada de uma empresa de aluguéis de hospedagem online através da plataforma "Kaggle". Com os resultados obtidos de ambos os cenários, foi possível comparar e entender o impacto em custos e performances das equipes de dados quando utilizado uma nova forma de armazenamento.

Palavras Chave: Engenharia de dados; Modelagem de dados; performance; armazenamento moderno; consultas de dados

Performance of different data modeling approaches in modern storage architecture.

Abstract

To address a scenario of significant increases in data usage by companies, new data storage tools are emerging focusing on speed and efficiency. Consequently, concepts from Inmon and Kimball regarding data modeling, dating back to the early 20th century and still widely applied today, become points of questioning. This is because the performance of these new technologies can lead to more performant data pipeline queries without the specific need for modeling. Therefore, the present study aims to analyze the behavior of a data query using two scenarios: the first one sourcing queries from normalized tables, and the other scenario querying directly from a non-normalized table, both using parquet files as the source. The dataset was obtained from an online rental company through the "Kaggle" platform. With the results obtained from both scenarios, it was possible to compare and understand the impact on costs and performance of data teams when using a new storage approach.

Keywords: Data engineering; data modeling; performance; modern storage architecture; data queries

Introdução

O estudo realizado pela "The State of Data Innovation" (2021), constatou que empresas maduras em dados e que tomam decisões baseadas neles, são mais inovadoras. Além disso, também deixa claro que, nas empresas líderes, "cloud analytics" deve ser de ponta para obter boa observância das informações que irão direcionar o negócio. Com isso, a necessidade por dados se torna cada vez mais constante para que todos possam utilizá-los tornando a cultura das empresas "data-driven".

Quando se observa o mercado, a tendência em empresas brasileiras é de um crescimento de 175% do volume de dados até 2025, saindo de 10 "petabytes [PB]" em 2020 e esperando 27 PB a serem processados em 2025 (Cappra Institute, 2020). Cada dia mais, é notória a real importância dos dados para as organizações, tornando ainda atual a frase de que os dados são o novo petróleo (HUMBY, 2006).

Conforme citado em “Cappra Institute” (2020), houve aumento da importância e da utilização de dados. Consequentemente, também ocorreram mudanças no cenário de armazenamento.

Antigamente as ferramentas eram desenvolvidas para armazenar dados através de arquivos em linhas, como formatos de bancos de dados relacionais tradicionais, como "MySQL", "Oracle" e "PostgreSQL", além de arquivos "Comma-Separated Values [CSV]" e "Javascript Object Notation [JSON]".

Atualmente, as ferramentas precisam entregar um conjunto performático: velocidade e baixo custo. Assim, o novo modo de armazenamento de dados, arquivos "Parquet", torna significativamente mais rápido o tempo de processamento (em 34 vezes) e reduz o espaço de armazenagem dos dados (em 87%). Isso se deve ao modo como se armazena a informação, salvando seus metadados com a versão, "schema", tipo e outras especificidades (Databricks, 2020).

No momento em que são requisitados os dados, espera-se que a leitura seja feita primeiramente pelos metadados, de modo a entregar somente o resultado requisitado pelo usuário sem precisar acessar todas as informações armazenadas (Apache Parquet, 2022).

Contudo, novas pesquisas surgem para entender se o método de armazenamento "parquet" é performático, como investigado por Duniam et al. (2022). Seu estudo sobre armazenamento de dados astronômicos demonstrou que tabelas em formato "parquet" podem oferecer uma alternativa viável ao armazenamento de tabelas em "Flexible Image

Transport System [FITS]", melhorando a performance em armazenamento de arquivos utilizados em escala. O comparativo do estudo foi feito com arquivos armazenados em FITS.

Quanto à modelagem de dados, os primeiros documentos direcionais pensando em consistência da informação para o negócio apareceram reforçando a necessidade de criação de tabelas com modelos relacionais fortes, escrito por Inmon, conhecido como "pai do armazenamento de dados" (Inmon, 2005). Para Kimball, a transformação dos dados extraídos devem ser fácil e rápida para que os mesmos sejam utilizados. Por isso, explana em sua literatura sobre a necessidade de modelagem dimensional através das construções de tabelas sobre o modelo de "star schema", a fim de obter uma melhor performance através de suas respectivas tabelas dimensionadas (Kimball, 2002).

As primeiras obras de Inmon são de 1970, enquanto as primeiras obras de Kimball, são datadas entre final do século vinte e início do século vinte e um. Sendo assim, deve-se considerar que todas essas definições estão ao redor do cenário da época em que as obras foram escritas, sendo o cenário composto pela maioria das ferramentas orientadas ao armazenamento de dados por linha ("Data Row Storage") e a um custoso - e devagar - espaço de armazenamento ("Expensive and Slow Storages").

Tendo em vista estes cenários de modelagem de dados, ao analisar os resultados da desnormalização de arquivos e seu desempenho nos formatos CSV e Parquet, constatou-se que a desnormalização das tabelas CSV resultou em um aumento significativo no tempo de execução, excedendo em mais de 3,5 vezes o período necessário para completar a consulta. Por outro lado, a desnormalização dos dados no formato Parquet revelou dois cenários importantes em relação à performance: em um, houve um aumento na eficiência de 3,2 vezes, enquanto no outro, não houve impacto na performance (Melo Silva, 2021).

Os novos processos de armazenamento se tornaram bastante conhecidos, porém, pouco se investiga sobre impactos relacionados à estruturação dos dados na performance das tabelas nestes novos métodos de armazenagem de dados, como citado e investigado por Costa (2017) e por Melo Silva (2021).

Dado o cenário exposto e as necessidades atuais, o presente trabalho tem como objetivo testar a performance das tabelas resultantes da "pipeline" de dados, tanto as que utilizam modelagem de dados conforme proposto por Kimball, quanto as que não utilizam. Sendo assim, a principal busca está relacionada ao quão performático é o cenário de "pipeline's" normalizadas "versus" não normalizadas em ambientes que utilizam armazenamento colunar em "parquet".

Material e Métodos

O método de pesquisa empregado será exploratório, buscando resultados quali-quanti (Gil, 2008). A pesquisa será estruturada em torno dos conceitos de normalização e desnormalização de bancos de dados. Quando um banco de dados está normalizado, isso significa que os dados foram organizados em tabelas separadas, seguindo regras de normalização que incluem a aplicação das formas normais, a definição de chaves primárias e estrangeiras, bem como o estabelecimento de relacionamentos entre tabelas. Por outro lado, a desnormalização de um banco de dados envolve a combinação de várias tabelas em uma única tabela, eliminando a estrutura normalizada.

A principal ferramenta utilizada será o terminal do computador, acompanhado da instalação do sistema "DuckDB", um sistema de gerenciamento de banco de dados que desempenhou um papel central para a análise de dados, desenhado para auxiliar e facilitar o desenvolvimento para usuários de equipes de dados.

Para executar as consultas, criar novas tabelas, fazer modificações e estabelecer a linhagem de dados, no banco de dados, a linguagem utilizada será "Structured Query Language [SQL]", sendo amplamente reconhecido como a linguagem de escolha para manipular dados em bancos de dados de pequeno ou grande porte, tornando as operações mais eficazes e eficientes.

Sendo assim, o primeiro cenário de análise, será através da requisição resultante da linhagem de dados exposta na Figura 1 abaixo, na qual, trará tabelas normalizadas como referência, através de dimensões, antecedentes a consulta realizada. Ressalta-se que todas as fontes serão de arquivos "parquet".

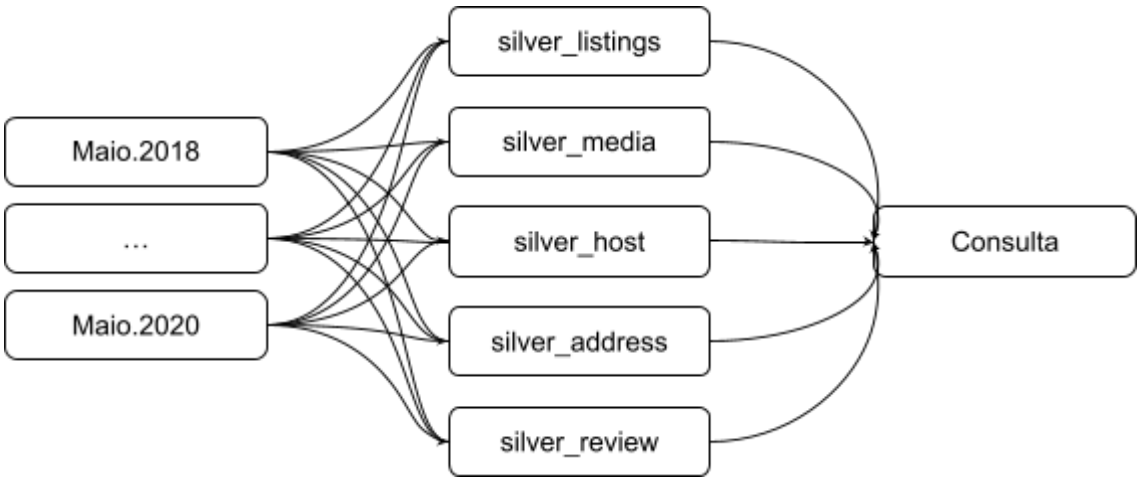

Figura 1. Representação do cenário 1, através da linhagem de dados até a consulta
Fonte: Dados originais da pesquisa

Para as tabelas descritas com as nomenclaturas iniciais "silver", na Figura 1, e criadas a partir do "dataset", especificados na tabela 1, temos como base o conceito "star schema", que possui como ideia principal separar os dados em duas partes: tabelas de fatos e tabelas dimensões. A principal diferença entre essas tabelas separadas se encontra que na tabela fato conterá medidas, enquanto as tabelas dimensão conterão atributos de descrição. Na Figura 2, a seguir, temos simplificadamente o relacionamento dessas tabelas.

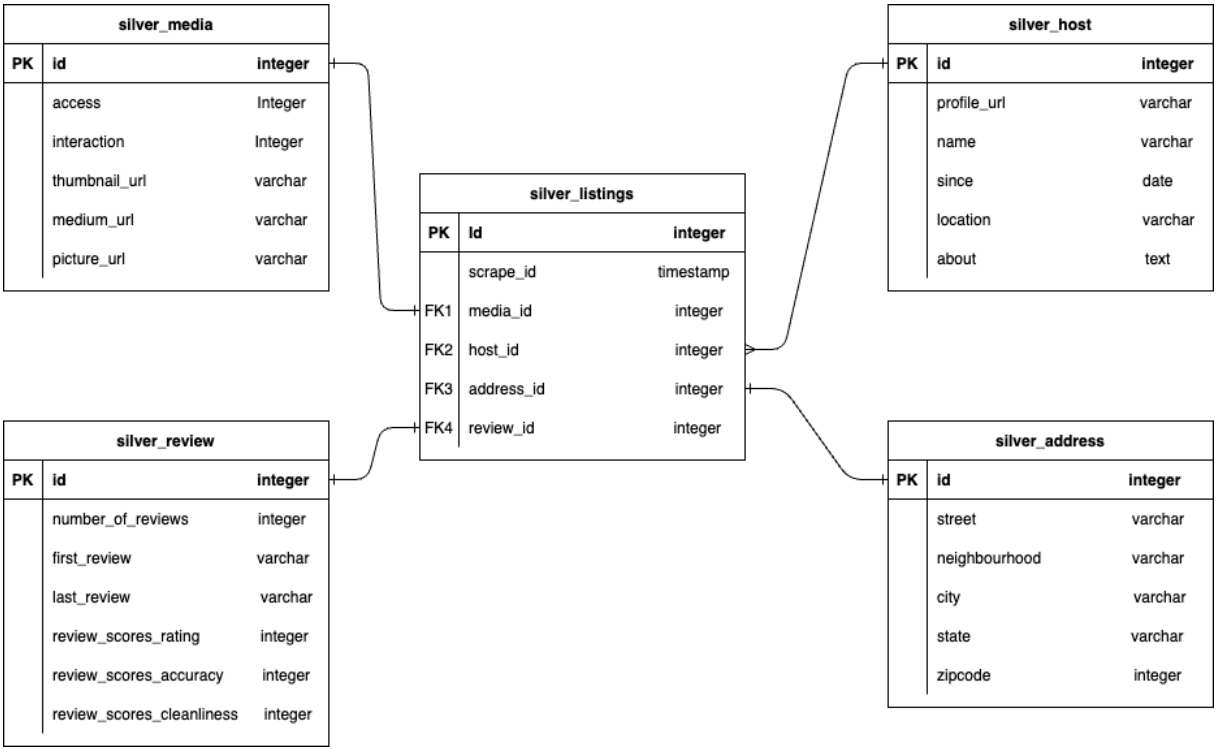

Figura 2. Representação da modelagem de dados das tabelas normalizadas do cenário 1
Fonte: Dados originais da pesquisa

O segundo cenário de análise, será através da consulta em tabela desnormalizada, que não utilizará dimensões e também utilizará do mesmo tipo de arquivo em seu armazenamento, neste caso, arquivos "parquet", conforme a Figura 3 abaixo.

Figura 3. Representação do cenário 2, através da linhagem de dados até a consulta
Fonte: Dados originais da pesquisa

Através dos dois cenários, deve-se requistar através da consulta ("query") informações do conjunto de dados analisados. A partir deste resultado, mede-se a performance desta consulta, através da utilização do comando "Explain Analyze", que mostrará o tempo e o plano de execução da consulta.

Vale ressaltar que, em ambos os cenários, apesar dos arquivos serem coletados em formato "CSV", serão transformados e utilizados em formato "parquet" para o "dataset" do presente estudo. Para realizar a transformação dos arquivos, utiliza-se da função "copy", função frequentemente utilizada em bancos de dados para exportar os dados de uma tabela para um arquivo externo em um determinado formato, sinalizando o tipo de arquivo desejado, que neste caso é o "parquet". Dessa forma, os arquivos são convertidos e podem ser utilizados através do novo formato.

"Dataset"

O "dataset" utilizado se refere à listagens de imóveis do Rio de Janeiro, que estavam ativas e foram coletadas mensalmente no "website" de uma plataforma de aluguéis de hospedagem mundialmente conhecida, do período de abril de 2018 a maio de 2020, com exceção do mês de Junho de 2018. Os dados foram coletados mensalmente através da disponibilização da ferramenta "Kaggle" e possuem características em "megabytes [MB]" conforme tabela 1 abaixo. ("Kaggle", 2023)

Tabela 1. Características das fontes de dados

Tabela	Arquivo armazenado em "CSV" em (MB)	Arquivo armazenado em "Parquet" em (MB)
abril2018	123,60	51,10
maio2018	123,10	50,80
julho2018	121,50	49,90
agosto2018	119,60	49,20
setembro2018	117,00	48,80
outubro2018	114,60	47,20
novembro2018	112,80	46,30
dezembro2018	114,00	46,70
janeiro2019	114,90	46,90
fevereiro2019	116,80	47,60

Tabela 1. Características das fontes de dados

Tabela	Arquivo armazenado em "CSV" em (MB)	Arquivo armazenado em "Parquet" em (MB)
marco2019	118,10	48,10
abril2019	117,50	47,80
maio2019	117,10	47,60
junho2019	116,70	47,60
julho2019	115,90	47,60
agosto2019	114,60	46,80
setembro2019	112,90	46,00
outubro2019	111,50	45,50
novembro2019	111,80	45,70
dezembro2019	114,90	46,70
janeiro2020	115,10	47,20
fevereiro2020	119,90	49,00
marco2020	118,50	48,40
abril2020	117,30	48,00
maio2020	116,70	47,80

Fonte: Dados originais da pesquisa

As escolhas dessas tabelas são justificadas pois, quando unidas, representarão uma quantidade de linhas e de armazenamento ocupado significativas para o efeito de comparação que este trabalho deseja realizar. Para que assim seja possível ser visualizada a diferença das performances entre os cenários estudados tanto em consulta quanto em espaço de arquivo armazenada. Além disso, com o "dataset" escolhido é possível efetuar a normalização do conteúdo em tabelas dimensionadas para obter o padrão esperado no primeiro cenário.

Características das tabelas normalizadas

Quanto às tabelas normalizadas, em dimensões, formadas a partir do "dataset" citado, e representadas pelo "star schema" da Figura 2, possuem as características relatadas na Tabela 2.

Tabela 2. Caracteristicas das tabelas normalizadas, dimensões

Tabela	Arquivo armazenado em "CSV" (MB)	Arquivo armazenado em "Parquet" (MB)
silver_listings	1.190,00	64,90
silver_address	172,00	5,90
silver_host	22,70	7,70
silver_media	652,20	31,00
silver_review	83,30	8,20

Fonte: Dados originais da pesquisa

Características da tabela desnormalizada

Para a tabela desnormalizada, tem-se as características relatadas na tabela 3 abaixo.

Tabela 3. Característica da tabela desnormalizada

Tabela	Arquivo armazenado em "CSV" (MB)	Arquivo armazenado em "Parquet" (MB)
bronze_listings	2.920,00	505,90

Fonte: Dados originais da pesquisa

Características das consultas utilizadas

Quanto às consultas realizadas para a análise de performance, estas possuem as seguintes descrições:

- Cenário 1: "Query" consultando fontes de arquivos "parquet`s" através das tabelas normalizadas

Neste cenário, é calculada a contagem de anúncios únicos, a contagem de URLs de fotos únicas, a contagem única de nomes de anfitriões, a contagem de nomes únicos de bairros e a média das notas de avaliação das hospedagens. A consulta utiliza a operação de junção a esquerda ("LEFT JOIN") em quatro tabelas diferentes ("silver_media","silver_host", "silver_address", "silver_review") e inclui uma subconsulta para calcular a média das notas de avaliação. Os resultados são filtrados para incluir apenas registros com ID de data de coleta correspondente a '20190313042552' e '20180414160018'. A consulta final é agrupada e ordenada pelo ID, "primary-key" da tabela fato.

- Cenário 2: "Query" consultando fontes de arquivos "parquet`s" através da tabela desnormalizada

Neste cenário, é calculada a contagem de anúncios únicos, a contagem de URLs de fotos únicas, a contagem única de nomes de anfitriões, a contagem de nomes únicos de bairros e a média das notas de avaliação das hospedagens. A consulta utiliza uma única tabela como fonte de dados, sem junções com o operador "left joins". Os resultados são filtrados para incluir apenas registros com ID de data de coleta correspondente a '20190313042552' e '20180414160018'. Em seguida, a consulta final é agrupada e ordenada, em ordem crescente, pelo ID, "primary-key" da tabela principal.

Resultados e Discussão

Observando a tabela 1, de materiais e métodos, existe um impacto notório na compração dos tamanhos dos dados armazenados entre arquivos "CSV" e "parquet`s". O conjunto do "dataset" em "parquet" representa uma redução de 60% do espaço ocupado.

De outro modo, para ser entendido o comportamento do "dataset" em seus respectivos cenários, parte-se para um número amostral: executa-se a solicitação cinco vezes, garantindo a limpeza de "caches", para obter o valor resultante de cada solicitação em cada cenário, conforme figura abaixo.

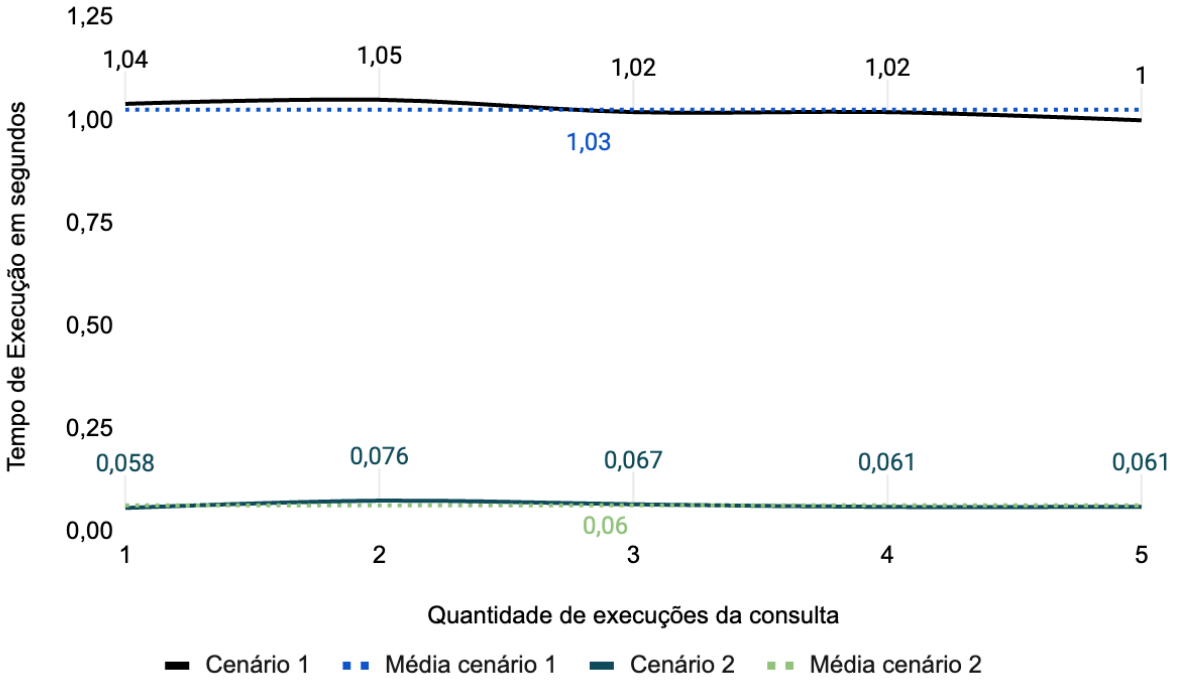

Figura 4. Resultado do tempo de execução de cada consulta realizada por cenários 1 e 2
Fonte: Resultados originais da pesquisa

Dessa forma, observando os dados obtidos na Figura 4, entende-se que o tempo de execução médio do cenário 1 é 1,03 segundos enquanto o do cenário 2 é 0,06 segundos. Olhando por escala, o cenário 2 é 16 vezes mais performático que o cenário 1.

Quando o plano de execução é analisado, observa-se a média de cada etapa em ambos os cenários. É importante destacar que o cenário 2 apresenta um plano de execução menor, pois não envolve várias junções para produzir o resultado da consulta.

Tabela 4. Análise da performance do plano de execução por cenários - não comparativos

Cenário 1		**Cenário 2**	
Execução	**Tempo (Em segundos)**	**Execução**	**Tempo (Em segundos)**
Order_by	0,00	Order_by	0,00
Hash_group_by	4,82	Hash_group_by	0,03
Projection	0,08	Projection	0,06
Filter	0,02	Filter	0,06
Hash_join	1,17	Read_parquet	0,27
Hash_group_by	0,02		
Projection	0,00		
Hash_join	0,05		
Read_parquet	0,03		
Filter	0,05		
Read_parquet	0,03		
Hash_join	0,69		
Read_parquet	0,02		
Hash_join	0,11		
Read_parquet	0,01		
Hash_join	0,12		
Read_parquet	0,04		
Filter	0,05		
Read_parquet	0,05		

Fonte: Resultados originais da pesquisa

Para os resultados apresentados acima, é necessário citar que a somatória do plano de execução não representa o total de tempo de execução da consulta, já que o valor representado em cada etapa é do total da operação individual, podendo existir paralelismo, "overhead" e imprecisões no cálculo total de cada operação.

Ao utilizar os resultados atuais como parâmetro multiplicativo para uma equipe de analistas de dados que lida com conjuntos de dados mais volumosos, é possível entender o impacto no tempo dedicado à investigação, o que pode resultar em melhorias significativas na produtividade dessa equipe e no desempenho financeiro da instituição.

Utilizando com base o salário mais baixo de um analista de dados senior, segundo o estudo "State of Data Brasil" (2022), de 8 mil reais, sem considerar impostos e outros fatores, foi investigado um problema e realizada uma consulta em um banco de dados de uma determinada hipótese por 30 minutos. No cenário 1, ao dividir o salário mensal desse analista pelo número médio de horas de trabalho em um mês, supondo que neste caso seja de 160 horas, tem-se o ganho por hora de aproximadamente 50 reais. Tendo em vista o cenário mencionado de 30 minutos, que representa 0,5 horas trabalhadas, tem-se que o tempo gasto gerará um custo para a empresa de aproximadamente 25 reais, referente a este funcionário executando a tarefa demandada. Já no cenário 2, utilizando da mesma base de cálculo citado acima, o custo financeiro para empresa equivale a menos de 2 reais deste mesmo funcionário, uma vez que o tempo gasto para executar a mesma consulta no banco de dados, levaria menos de 2 minutos.

Quando observa-se o impacto em uma equipe de 30 analistas, utilizando os mesmos parâmetros, no cenário 1 a empresa teria um custo de 750 reais. Enquanto no cenário 2, o custo seria de 50 reais para executar a mesma operação. Dessa forma, pode-se concluir que a economia financeira gerada para a empresa é de mais de 90%. Além disso, ressalta-se que a empresa também se beneficia no aumento da produtividade dessa equipe.

Simulando o custo de armazenamento com base nos dados fornecidos pelo "Databricks" (2020), observa-se que para o armazenamento em formato CSV, o custo de 1 terabyte [TB] é de 5,75 dólares. A fonte utilizada neste estudo ocupa um espaço de 3 gigabytes [GB], resultando em um custo de 0,02 dólares. Por outro lado, quando convertida para o formato Parquet, a mesma fonte ocuparia apenas 1 gigabyte e gera um custo de menos 0,01 dólares. Isso representa uma economia de mais de 50% nos custos de armazenamento. É importante salientar que, embora a base de dados utilizada não seja extensa o suficiente para uma comparação significativa de armazenamento, ela proporciona uma oportunidade valiosa para a realização de simulações, visando o aprendizado.

Considerações Finais

É recomendável conduzir um estudo sobre os custos e tempo de exeucução para ambas as fontes com um banco de dados maior. Porém, os fatos apresentados, conclui que a performance de arquivos "Parquet" sem modelagem de dados, foi 16 vezes mais performática do que se tivesse modelagens, levando em média 0.065 segundos para executar, enquanto o cenário com modelagem, normalizado, retornou em 1.03 segundos.

Ao utilizar os resultados de performance de tempos de execução em ambos os cenários como parâmetro multiplicativo para uma equipe de analistas de dados que lida com conjuntos de dados mais volumosos, é possível concluir que pode ser otimizado o tempo dedicado à investigação e o custo, ao utilizar cenários de ambientes modernos sem modelagem, desnormalizados, o que pode resultar em melhorias significativas na produtividade e no desempenho financeiro da instituição. Individualmente, a diferença proporcional é significativa, porém, quando analisada olhando para uma equipe como um todo, se torna ainda mais relevante. Vale ressaltar, que existe também um impacto na redução de espaço no armazenamento de arquivos.

Desse modo, ideias e dialogos compostos na literatura e escritos no final do século XX, são questionados ao serem postos em prática, tendo em vista as novas arquiteturas de armazenamento existentes hoje, são refutadas teorias de Kimball e Inmon.

Ainda assim, com todas as evidências em questão, deve-se ressaltar que existe uma importância para as modelagens de dados ocorrerem, como em casos em que as empresas necessitem de tabelas documentadas e organizadas semanticamente para aumentar a qualidade de catalogação da informação, mantendo uma governança qualificada, tanto de acessos, quanto para gerir mudanças das informações existentes. A importância é significativamente observada, do ponto de vista de criações de modelos de inteligências artificiais e ciências de dados.

O presente estudo, não se limitou somente a análise do objetivo em questão, de analisar resultados e performance de arquivos parquet, modo de armazenamento de arquivos modernos, e apontou no apêndice, dois novos cenários utilizando fontes CSV, modo de armazenamento de arquivos não modernos, os comparando com cenários expostos no presente trabalho.

Agradecimento

Durante a vida, sempre estamos suscetíveis a sorte que traz oportunidades. Por isso, agradeço a Deus pelas sortes que se acumularam em minha vida e que auxiliaram

positivamente na minha trajetória até aqui e que hoje, me permitem ser um privilegiado. Status esse que ao longo da minha vida era muito distante. Por outro lado, agradeço a minha família, amigos e noiva que sempre estão próximos ajudando todos os dias para que eu consiga aproveitar ao máximo todas as oportunidades que aparecem. Dentre os amigos, devo ressaltar um agradecimento especial ao Rafael Olszewski, que assim como eu, também é apaixonado pelo tema e que me escutou e ajudou a desenvolver ideias para o presente projeto, baseadas em nossas vivências profissionais.

Referências

Cappra Institute. 2020. Maturidade análitica das organizações brasileiras Disponível em: <https://www.cappra.institute/ima> Acessado em 05 de novembro de 2022
HUMBY, Clive. Data is the new oil. Proc. ANA Sr. Marketer's Summit. Evanston, IL, USA, 2006.

Costa, E., Costa, C., and Santos, M. Y. 2017. Efficient big data modelling and organization for hadoop hive-based data warehouses. In Themistocleous. European, Mediterranean and Middle Eastern Conference on Information Systems. Springer International Publishing.

Databricks. What is parquet?. Disponível em: <https://www.databricks.com/glossary/what-is-parquet> Acessado em 06 de novembro de 2022

Duniam, G.; Kitaeffk, V.V.; Wicenec, A. 2022. Data modelling approaches to astronomical data: Mapping large spectral line data cubes to dimensional data models, Astronomy and Computing, Volume 38.

Duniam, G. 2017. Big Data Architecture in Radio Astronomy: The effectiveness of the Hadoop/Hive/Spark ecosystem in data analysis of large astronomical data collections.

Gil, Antonio Carlos. 2008. Como elaborar projetos de pesquisa. 4. ed. São Paulo: Atlas.

Inmon, W. H. 2005, Building the Data Warehouse, 4º Edition. Wiley Publishing, Inc.

Kimball, R.; Ross, M. 2002. The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. New York. John Wiley & Sons.

Kaggle. Rio de Janeiro Airbnb. Disponível em: <https://www.kaggle.com/datasets/allanbruno/airbnb-rio-de-janeiro> Acessado em 20 de agosto de 2023

Melo Silva, Felipe. 2021, Avaliação da Utilização de Arquivos Desnormalizados no Spark SQL. Programa de Graduação em Engenharia de Computação do Centro de Informática. Universidade Federal de Pernambuco

State of Data Brasil. . Disponível em: <https://www.stateofdata.com.br/> Acessado em 25 de Janeiro de 2023

Splunk. 2021. The State of Data Innovation. Disponível em: <https://www.splunk.com/en_us/form/state-of-data-innovation.html> Acessado em 07 de novembro de 2022

Apêndice

Ao analisar, separadamente, o resultado performático de arquivos "CSV", considerados na presente pesquisa como um modelo de armazenamento não moderno, obteve-se os seguintes resultados médios de performances considerando o cenário 3 e o cenário 4 descritos:

- Cenário 3: "Query" consultando fontes de arquivos "CSV" através das tabelas normalizados

Neste cenário, replica-se a consulta de cenário 1, modificando apenas a fonte de arquivo armazenado, e é calculada a contagem de anúncios únicos, a contagem de URLs de fotos únicas, a contagem única de nomes de anfitriões, a contagem de nomes únicos de bairros e a média das notas de avaliação das hospedagens. Essa consulta utiliza a operação de junção a esquerda ("LEFT JOIN") em quatro tabelas diferentes ("silver_media","silver_host", "silver_address", "silver_review") e inclui uma subconsulta para calcular a média das notas de avaliação. Os resultados são filtrados para incluir apenas registros com ID de data de coleta correspondente a '20190313042552' e '20180414160018'. A consulta final é agrupada e ordenada pelo ID, "primary-key" da tabela fato.

- Cenário 4: "Query" consultando fontes de arquivos "CSV" através da tabela desnormalizados

Neste cenário, replica-se a consulta de cenário 2, modificando apenas a fonte de arquivo armazenado, e é calculada a contagem de anúncios únicos, a contagem de URLs de fotos únicas, a contagem única de nomes de anfitriões, a contagem de nomes únicos de bairros e a média das notas de avaliação das hospedagens. A consulta utiliza uma única tabela como fonte de dados, sem junções com o operador "left joins". Os resultados são filtrados para incluir apenas registros com ID de data de coleta correspondente a '20190313042552' e '20180414160018'. Em seguida, a consulta final é agrupada e ordenada, em ordem crescente, pelo ID, "primary-key" da tabela principal.

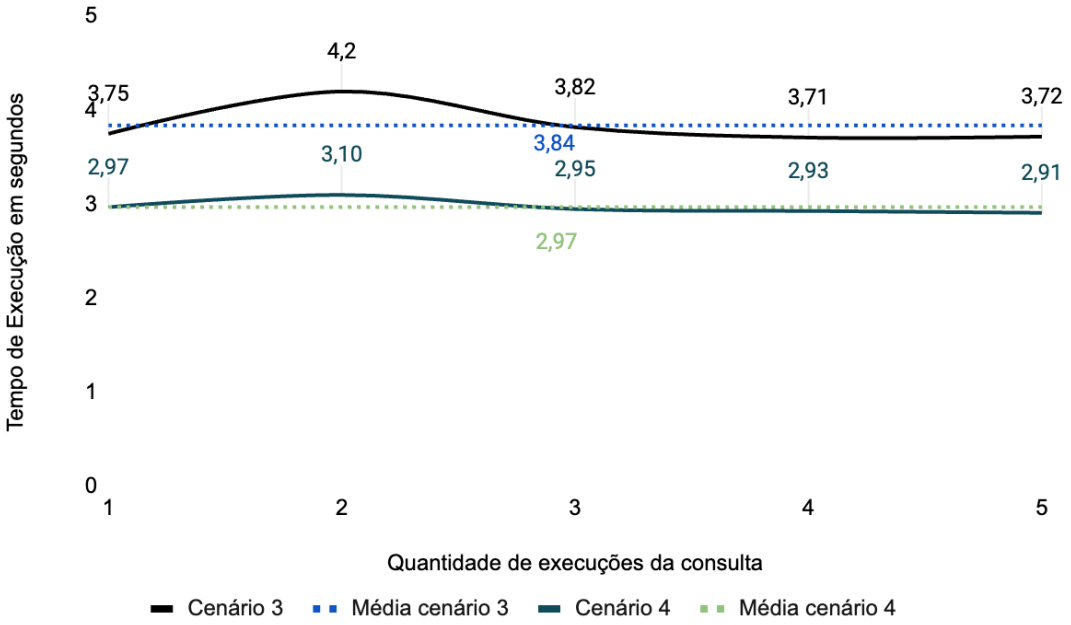

Figura 5. Resultado do tempo de execução de cada consulta realizada por cenários 3 e 4
Fonte: Resultados originais da pesquisa

Com os resultados obtidos, conforme a Figura 5, a média do tempo de execução do cenário 3 foi de 3,84 segundos, contra 2,97 segundos do cenário 4. No que diz respeito à comparação entre os resultados do cenário 1 e do cenário 2, observamos uma diferença notável. Por outro lado, ao comparar o cenário 3 com o cenário 4, notamos que a diferença é mais modesta, sendo apenas uma vez menos performático.

Porém, quando é comparado o cenário 1 com o cenário 3 e o cenário 1 com o cenário 4, temos uma performance 3 vezes maior. Observando o cenário 2 com o cenário 4, tem-se a maior performance, mostrando que um arquivo "parquet" sem modelagem, desnormalizado, é 49 vezes mais performático que um arquivo "csv" desnormalizado.

Os paragráfos acima citados, demonstram que a decisão de qual modelagem será aplicada, dependerá do formato de arquivos que serão utilizados, além de decisões baseadas na facilidade de compreender cada tipo de modelagem e dos dados a serem trabalhados.