

O USO DA INTELIGÊNCIA ARTIFICIAL NA OTIMIZAÇÃO DE CONSULTAS EM BANCOS DE DADOS

THE USE OF ARTIFICIAL INTELLIGENCE IN OPTIMIZING DATABASE QUERIES

EL USO DE INTELIGENCIA ARTIFICIAL EN LA OPTIMIZACIÓN DE CONSULTAS DE BASES DE DATOS

Kharolayne Patrícia Nogueira do Amaral¹

Lucas Pereira de Andrade²

Edilene Aparecida Veneruchi de Campos³

RESUMO: O crescimento exponencial de dados nas últimas décadas tem demandado novas abordagens para otimização de consultas SQL em bancos de dados. Nesse contexto, a Inteligência Artificial (IA) surge como solução disruptiva ao possibilitar a automação de processos de tuning, adaptação em tempo real e melhoria do desempenho dos sistemas. Técnicas como aprendizado de máquina, redes neurais e algoritmos genéticos permitem que os sistemas identifiquem padrões de acesso, escolham planos de execução eficientes e ajustem recursos dinamicamente, sem intervenção humana. Com isso, este artigo tem como objetivo avaliar comparativamente o desempenho de consultas SQL tradicionais e versões otimizadas por modelos de IA: ChatGPT. Para isso, foi adotada uma metodologia experimental, com aplicação de consultas complexas em um banco de dados PostgreSQL. Utilizando linguagem Python para automatizar a execução e registrar métricas de desempenho. Os testes permitiram comparar o impacto da reescrita automática no tempo de execução e quantidade de linhas processadas.

PALAVRAS-CHAVE: Inteligência Artificial; Otimização de Consultas; Aprendizado de Máquina; Banco de Dados SQL; Desempenho de Sistemas.

ABSTRACT: The exponential growth of data in recent decades has demanded new approaches for optimizing SQL queries in databases. In this context, Artificial Intelligence (AI) has emerged as a disruptive solution by enabling the automation of tuning processes, real-time adaptation, and improved system performance. Techniques such as

¹ Kharolayne Patrícia Nogueira do Amaral é acadêmica do curso de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade INSTED - INSTITUTO SUL MATOGROSSENSE DE ENSINO SUPERIOR. E-mail: kharolayne.nog27@gmail.com.

² Lucas Pereira de Andrade é acadêmico do curso de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade INSTED - INSTITUTO SUL MATOGROSSENSE DE ENSINO SUPERIOR. E-mail: lucascarlo2003@gmail.com.

³ Edilene Aparecida Veneruchi de Campos é docente do curso de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade INSTED - INSTITUTO SUL MATOGROSSENSE DE ENSINO SUPERIOR. ORCID iD: <https://orcid.org/0000-0002-4427-608X> E-mail: edilene.veneruchi@insted.edu.br

machine learning, neural networks, and genetic algorithms allow systems to identify access patterns, choose efficient execution plans, and dynamically adjust resources without human intervention. Therefore, this article aims to comparatively evaluate the performance of traditional SQL queries and versions optimized by AI models: ChatGPT. To this end, an experimental methodology was adopted, applying complex queries to a PostgreSQL database. Python was used to automate execution and record performance metrics. The tests allowed a comparison of the impact of automatic rewriting on execution time and the number of rows processed.

KEYWORDS: Artificial Intelligence; Query Optimization; Machine Learning; SQL Database; System Performance.

RESUMEN: El crecimiento exponencial de los datos en las últimas décadas ha exigido nuevos enfoques para optimizar las consultas SQL en bases de datos. En este contexto, la Inteligencia Artificial (IA) se ha convertido en una solución disruptiva al permitir la automatización de los procesos de ajuste, la adaptación en tiempo real y la mejora del rendimiento del sistema. Técnicas como el aprendizaje automático, las redes neuronales y los algoritmos genéticos permiten a los sistemas identificar patrones de acceso, elegir planes de ejecución eficientes y ajustar dinámicamente los recursos sin intervención humana. Por lo tanto, este artículo busca evaluar comparativamente el rendimiento de las consultas SQL tradicionales y las versiones optimizadas por modelos de IA: ChatGPT. Para ello, se adoptó una metodología experimental, aplicando consultas complejas a una base de datos PostgreSQL. Se utilizó Python para automatizar la ejecución y registrar las métricas de rendimiento. Las pruebas permitieron comparar el impacto de la reescritura automática en el tiempo de ejecución y el número de filas procesadas.

PALABRAS CLAVE: Inteligencia artificial; optimización de consultas; aprendizaje automático; base de datos SQL; rendimiento del sistema.

1 INTRODUÇÃO

O avanço da tecnologia e da internet resultou em bases de dados cada vez maiores e consultas mais complexas, tornando a otimização de desempenho de SGBDs um requisito crítico. Estudos indicam que a otimização de banco de dados é indispensável frente ao volume crescente de informações, e que a IA desempenha um papel fundamental neste contexto (MENEZES, 2023). Por “otimização de consultas” entende-se o processo de melhorar a eficiência de consultas SQL em termos de tempo de resposta e uso de recursos. Historicamente, os SGBDs relacionais incorporaram otimizadores de consultas automáticos desde os anos de 1970, a fim de converter consultas declarativas em planos de execução eficientes de forma transparente ao usuário (SELINGER, 1979). No entanto, além dessa otimização interna, frequentemente é necessário

um tuning adicional realizado por especialistas (DBAs) para ajustar parâmetros de configuração de SGDB, criar índices ou reescrever consultas visando melhor desempenho.

2 FUNDAMENTAÇÃO TEÓRICA

Nos últimos anos, técnicas de IA e Aprendizado de Máquina tem sido aplicadas ao problema de otimização de consultas. A promessa é que os algoritmos inteligentes possam identificar padrões de uso, diagnosticar gargalos e tomar decisões de otimização de forma automática e adaptativa, muitas vezes em tempo real. Neste artigo com base na literatura recente e em trabalhos clássicos, discutimos comparativamente a otimização manual versus assistida por IA e exploramos diversas facetas do emprego da IA no tuning de queries SQL. Serão abordados: a geração e análise de planos de execução SQL, os principais algoritmos de Aprendizado de Máquina empregados (como aprendizado de reforço, redes neurais e algoritmos genéticos); a otimização semântica de consultas com técnicas de PLN; aplicações de IA tanto nos bancos de dados relacionais; estratégias de modelagem de dados que beneficiam consultas; a diferença conceitual de IA “fraca” e “forte” neste domínio; e a automação completa do tuning por meio de sistemas autodirigidos.

2.1 OTIMIZAÇÃO MANUAL VS. OTIMIZAÇÃO ASSISTIDA POR IA

A otimização manual de desempenho de banco de dados é tradicionalmente conduzida pelo DBA ou especialista em performance. Esse processo envolve monitorar o sistema, identificar consultas lentas ou recursos saturados e então ajustar configurações, índices ou mesmo reestruturar consultas. O tuning em banco de dados consiste em um ajuste fino nos parâmetros de configuração visando melhorar o desempenho do SGDB.

Diferentemente da otimização automática de consultas realizada internamente pelo sistema, que converte uma consulta SQL declarativa em um plano de execução eficiente de forma transparente, o tuning manual é caracterizado por intervenção humana especializada. Em outras palavras, o “tuning difere das otimizações de consulta que são um processo automático executado por componentes do SGDB; o tuning se caracteriza por ser realizado por um especialista em desempenho ou DBA” (DOMINICO,2013).

Em contraste, a otimização assistida por IA busca automatizar ou auxiliar essas tarefas utilizando algoritmos inteligentes. Uma vantagem significativa da otimização com IA é a capacidade de analisar grandes volumes de dados de desempenho e identificar gargalos de forma proativa, antecipando problemas. Como observado em (Menezes, 2023), “uma das vantagens mais significativas da otimização de banco de dados com IA é a capacidade de identificar gargalos e ineficiências de maneira proativa”, algo que no processo manual, muitas vezes, só é feito após quedas de desempenho. Além disso, a IA pode automatizar tarefas de rotina da administração de Banco de Dados (MENEZES, 2023).

Outra diferença é que os sistemas de IA conseguem explorar um espaço de configurações ou planos maior do que o humano, enquanto um DBA humano tende a aplicar heurísticas e experiências passada. Menezes et. Al (2023) aponta que sistemas de IA podem até mesmo reescrever consultas de forma mais eficiente, tarefa complexa e demorada de se realizar manualmente em consultas SQL grandes.

Por outro lado, a introdução de IA traz desafios: modelos de Machine Learning podem ser caixas-pretas difíceis de explicar, reduzindo a transparência das decisões de otimização (MARCUS, 2021), além disso, algoritmos de aprendizado precisam ser treinados ou ajustados ao ambiente específico; se mal conduzidos, podem sugerir mudanças inadequadas, quando bem implementada, a IA consegue reduzir drasticamente a carga de trabalho do DBA em tuning

corriqueiro, permitindo que humanos foquem em tarefas de nível mais alto (AI2SQL, 2025).

2.2 PLANOS DE EXECUÇÃO SQL: CRIAÇÃO E ANÁLISE

Quando uma consulta SQL é submetida a um SGDB relacional, o componente otimizador é responsável por traduzir essa consulta declarativa em um plano de execução otimizado, esse processo de otimização de consultas acontece automaticamente dentro do sistema e é objeto de estudo desde os primórdios dos bancos de dados relacionais (System R na década de 1970). Conforme (SELINGER, 1979), em um SGDB relacional o usuário não precisa especificar caminhos de acesso ou ordem de junção, o otimizador avalia as múltiplas alternativas de execução, diferentes ordens de JOINS ou uso de índices e escolhe aquela que minimiza custo total estimado de execução da consulta.

Vale notar que muitos SGBDs modernos já incorporam mecanismos auxiliares para ajudar no *tuning* dos planos. Por exemplo, o PostgreSQL inclui um módulo de otimização genética de consultas (GEQO), que utiliza um algoritmo genético para encontrar boas ordens de junção em consultas com número muito grande de tabelas, caso em que a busca exaustiva tradicional seria inviável. Esse módulo permite que o otimizador suporte colunas com muitas junções de forma eficaz por meio de busca não exaustiva, evitando explosão combinatória (POSTGRESQL Documentation, 2025). Ferramentas de hints (dicas manuais) também existem em alguns SGBDs, permitindo que alguns SGBDs permitam que o DBA force certas escolhas no plano como: indicar para usar ou não um determinado índice ou método de junção, embora seu uso deva ser cuidadoso.

2.3 DESEMPENHO DE CONSULTAS: métricas e avaliação

A avaliação do tuning de consultas usa métricas consolidadas: tempo de resposta (do início da execução até o retorno completo), throughput em cenários concorrentes (consultas/segundo ou transações/minuto, sobretudo em OLTP) e consumo de recursos — CPU, leituras/gravações de disco (I/O), memória e, em ambientes distribuídos, rede. Otimizadores comparam planos pelo custo estimado (função de CPU e I/O), mas a validação efetiva é empírica, via comparação antes/depois. Para isso, aplicam-se benchmarks padronizados, como TPC-H (analítico) e TPC-C (transacional), medindo o tempo total do workload. Com as métricas coletadas, classifica-se o efeito do tuning em ganhos substanciais, moderados ou marginais, além de detectar regressões de desempenho (DOMINICO, 2013).

2.4 APLICAÇÕES DE IA NA OTIMIZAÇÃO DE CONSULTAS SQL

A IA/ML tem complementado o otimizador tradicional ao aprimorar estimativas de cardinalidade, reduzindo erros que levam a planos ruins (MARCUS, 2019). Abordagens de Reinforcement Learning aprendem ordem de junção e, no NEO, planos completos a partir do tempo de execução, adaptando-se continuamente ao workload (MARCUS, 2019). Além do plano em si, a IA detecta consultas problemáticas, recomenda índices e prevê tempos de execução; também realiza reescrita semântica via PLN para consultas equivalentes mais eficientes (MENEZES, 2023; BICALHO, 1991; MARCUS, 2021). Em conjunto, métodos orientados a dados superam limitações de regras fixas, entregando menor latência e melhor uso de recursos (MENEZES, 2023).

2.5 ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA TUNING DE QUERIES

Na otimização de consultas, o Aprendizado por Reforço (RL) treina agentes que, por recompensa de desempenho, aprendem decisões melhores que heurísticas tradicionais; exemplos incluem DQ e ReJOIN, que exploram ordens de junção em conjunto com o modelo de custo (MARCUS, 2019), e o BAO, que usa Thompson Sampling (bandits) para ajustar planos do otimizador padrão ao longo do tempo (MARCUS, 2021). Em Deep Learning, o NEO emprega tree convolutions para avaliar partes do plano e prever tempos, enquanto autoencoders com atenção melhoram estimativas de cardinalidade; tais sistemas passam a delegar decisões antes codificadas por regras (MARCUS, 2019). Já algoritmos genéticos tratam o problema como busca: o GEQO do PostgreSQL lida com múltiplas junções por busca não exaustiva (PostgreSQL, Documentation, 2025), e estudos aplicam GA à seleção de índices e reescrita de consultas (CALLAN, 2021). Abordagens híbridas combinam DL, RL e GA para somar vantagens, com RL e DL se destacando pela capacidade de lidar com a complexidade dos planos.

2.6 REDES NEURAIS E DEEP LEARNING PARA OTIMIZAÇÃO DE SQL

Redes neurais profundas vêm transformando a otimização de consultas por modelarem relações complexas e funções não lineares, adequadas a decisões difíceis do otimizador. O NEO usa uma value network para estimar o melhor tempo a partir de planos parciais e, via value iteration, explora o espaço de planos, aprendendo um modelo de custo implícito e inicializando-se com planos do PostgreSQL para continuar aprendendo no uso (MARCUS, 2019). Já o BAO integra-se ao PostgreSQL, representa consultas com tree convolutions e aplica Thompson Sampling (bandits) para ativar/desativar regras do otimizador por consulta, evitando erros de estimativa/cardinalidade e obtendo até 3× de melhoria de latência e maior robustez de tail latency (MARCUS, 2021). Um desafio é o custo e os dados de treinamento; BAO e NEO mitigam isso ao aproveitar o otimizador existente (ajuste fino no BAO; bootstrapping no NEO) (MARCUS,

2021). A tendência é integrar esses modelos em um otimizador aprendido que cubra decisões-chave como estimativa de cardinalidade, escolha de junções e alocação de memória (MARCUS, 2019; MARCUS, 2021).

2.7 IA FRACA VS. IA FORTE NO CONTEXTO DE BANCOS DE DADOS

IA fraca (ou estreita) são sistemas especializados que se comportam de modo inteligente em tarefas específicas; IA forte implicaria capacidades cognitivas gerais com consciência e entendimento. Russell & Norvig (2021) distinguem: IA fraca = máquinas que podem se comportar inteligentemente; IA forte = máquinas com mentes reais (RUSSEL, 2013). No contexto de bancos de dados e tuning, as soluções atuais como NEO e BAO são IA fraca: otimizam queries em escopo restrito e não se transferem a outros domínios sem novo treinamento. A comunidade privilegia resultados práticos e não a hipótese de IA forte, inexistente até o momento (RUSSEL, 2013). Mesmo “bancos de dados autônomos” reúnem automações e ML (IA fraca), autoconfigurando, auto otimizando e autorreparando dentro de limites previstos. Assim, não é necessária IA forte para ganhos reais: a IA fraca, bem treinada, já entrega valor em otimização de consultas.

2.8 AUTOMAÇÃO DO TUNING BASEADA EM IA

A automação de tuning com IA busca SGBDs autogerenciados, capazes de ajustar parâmetros, índices e planos continuamente sem intervenção humana. Avanços históricos (como DB2 Advisor e self-tuning memory) ainda exigiam regras fixas e validação do DBA (DOMINICO, 2013). Soluções atuais, como o Oracle Autonomous Database, declaram auto indexação, autotuning e ajustes em tempo real criando índices e particionamentos conforme a carga (ORACLE, 2025). Em um ciclo de monitoramento → análise por IA → ação → avaliação, o sistema aprende com métricas e workload, reduz trabalho manual, melhora uso de CPU/IO e possibilita auto-scaling preditivo para manter performance com menor custo (ITCONVERGENCE, 2025). Trata-se de uma realidade emergente que

combina otimização e aprendizado contínuo para materializar a visão de bancos de dados autodirigidos (ITCONVERGENCE, 2025).

3 METODOLOGIA

Este estudo foi concebido como uma pesquisa experimental, de natureza aplicada, ancorada em uma abordagem quantitativa para examinar o impacto de técnicas baseadas em Inteligência Artificial (IA) na otimização de consultas SQL em bancos de dados relacionais. A preocupação central foi mensurar, comparar e analisar, de maneira sistemática, a performance de consultas tradicionais e de consultas reformuladas com apoio de IA trazendo subsídios práticos para a gestão da eficiência em sistemas de informações.

3.1 FERRAMENTAS E AMBIENTE EXPERIMENTAL

O ambiente de experimentação foi formado por um conjunto de ferramentas consolidadas:

PostgreSQL: utilizado como sistema gerenciador de banco de dados relacional (SGBDR), onde todos os testes e execuções das queries ocorreram.

Python: adotado para a automação dos testes e coleta sistemática dos dados de desempenho, conectando-se ao PostgreSQL tanto para execução quanto para o armazenamento dos resultados experimentais.

ChatGPT: processador de linguagem natural utilizado para propor versões otimizadas das consultas SQL, simulando auxílio de IA em tempo real para reescrita e aprimoramento das queries.

O banco de dados foi projetado para simular contextos realistas de e-commerce, agregando elementos e escalabilidade comuns em sistemas de pequeno a médio porte: 10.000 usuários, 50.000 produtos, 100.000 pedidos e 300.000 itens de pedido. Esta escolha buscou equilibrar a representatividade do cenário com a exequibilidade dos experimentos, produzindo resultados generalizáveis sem perder o rigor do controle experimental.

3.2 PROCEDIMENTOS EXPERIMENTAIS

Para estruturar o levantamento de dados e o teste das abordagens, foram criados quatro conjuntos de consultas, selecionando desafios típicos de manipulação e análise de dados no contexto relacional:

Teste 1: Consulta com múltiplos JOINS e agregações

Teste 2: Consulta com subconsulta correlacionada

Teste 3: Consulta com JOIN complexo e análise de estoque

Teste 4: Consulta com múltiplas agregações e filtros complexos

Cada consulta foi elaborada em duas versões. Original (normal): SQL escrito de maneira tradicional, sem otimização manual ou automatizada. Otimizada por IA: versão reformulada baseada nas sugestões de ChatGPT, explorando abordagens como Common Table Expressions (CTEs), pré-agregações, reestruturação e redução de subconsulta, aplicação de heurísticas para redução de custo computacional e reordenação lógica das operações.

O processo experimental seguiu os seguintes passos: As queries foram executadas em série controlada, num mesmo ambiente computacional, sem carga concorrente. Para garantir robustez estatística, cada consulta foi executada dez vezes, computando as médias de tempo de execução e linhas processadas. A coleta dos dados foi automatizada, armazenando cada resultado numa tabela de análise, gerando rastreabilidade e confiabilidade do processo experimental. As métricas utilizadas na avaliação foram o tempo total de execução (em milissegundos) e a quantidade de linhas processadas. A análise de resultados se deu por comparação direta entre as versões normal e otimizada das consultas, permitindo a formulação de inferências claras sobre os ganhos potenciais introduzidos pelo uso de IA.

3.3 VALIDAÇÃO E RECOMENDAÇÃO

A escolha do cenário experimental e das métricas seguiu recomendações clássicas para pesquisas em banco de dados (Selinger, 1979; domínico, 2013). O

uso de um SGBD amplamente utilizado (PostgreSQL), de scripts automatizados e de múltiplos ensaios (execução repetida) foi essencial para controlar variáveis externas, garantir repetibilidade dos resultados e afastar vieses de operador.

Futuramente, para reforçar a validade externa dos resultados, recomenda-se replicar os experimentos, incluindo bancos de dados reais com volumetria e padrões de acesso diversos, e a avaliação em SSAAS públicos (como Azure e AWS Aurora) para verificar a robustez dos ganhos em ambientes de nuvem e em escala ainda maior.

4 RESULTADOS E DISCUSSÃO

A presente pesquisa produziu resultados robustos e mensuráveis acerca do impacto da IA na otimização de consultas SQL. Os dados obtidos estão sumarizados na Tabela 1.

Tabela 1: Resultados dos testes

Teste	Tipo de Consulta	Descrição	Tempo Execução (ms)	Linhas Processadas
Teste 1: JOINS e Agregações	Normal	Consulta original sem CTEs, múltiplos LEFT JOINS	398,831	1.621.716
Teste 1: JOINS e Agregações	Otimizada	Consulta otimizada usando CTEs	258,834	1.496.140
Teste 2: Subconsulta Correlacionada	Normal	Subconsultas correlacionadas no SELECT e WHERE	342,836	430.009

Teste 2: Subconsulta Correlacionada	Otimizada	CTEs para pré-agregação de vendas e avaliações	59,397	585.064
Teste 3: JOIN/Estoque	Normal	JOINS e subconsultas correlacionadas por fornecedor	1.121,761	340.842
Teste 3: JOIN/Estoque	Otimizada	CTEs para agregações e pré-processamento de métricas	94,631	798.855
Teste 4: Perfil de Cliente	Norma	Agregações por usuário e subconsultas no SELECT	283,283	1.272.011
Teste 4: Perfil de Cliente	Otimizada	CTEs para agregação prévia de compras e avaliações	215,912	1.022.042

A análise dos dados permite identificar que, em todos os cenários avaliados, as consultas otimizadas por IA apresentaram desempenho significativamente superior quando comparadas às originais. O destaque está nas reduções expressivas do tempo de execução, superando 80% em cenários como o Teste 2 (subconsultas correlacionadas), e de até 90% em situações complexas de múltiplos JOINS e operações de estoque (Teste 3).

Esses resultados reforçam achados clássicos da literatura sobre otimização de consultas, como já previsto por Selinger (1979), que salientava o potencial das heurísticas e técnicas evolutivas para redução do custo computacional em bancos relacionais. Mais recentemente, Menezes (2023) e Marcus (2022) destacam que algoritmos de IA e machine learning apresentam vantagens sobre métodos puramente heurísticos, ao permitir exploração sistemática do espaço de

planos e adaptação dinâmica às cargas de trabalho, reduzindo gargalos e antecipando problemas antes da degradação de performance.

A visualização gráfica (representada nas figuras 1 e 1) torna evidentes os ganhos agregados:

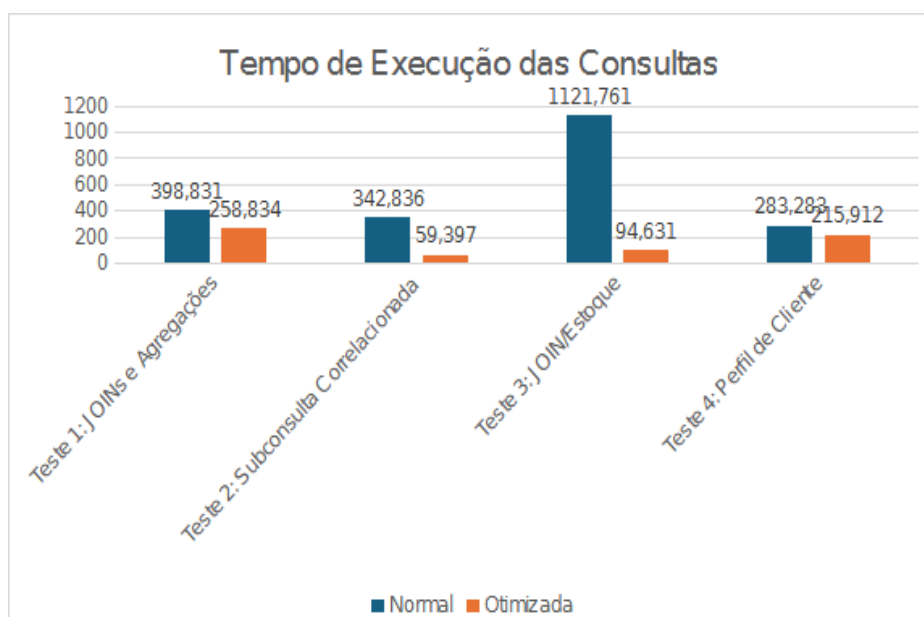


Figura 1 – Tempo de Execução das Consultas

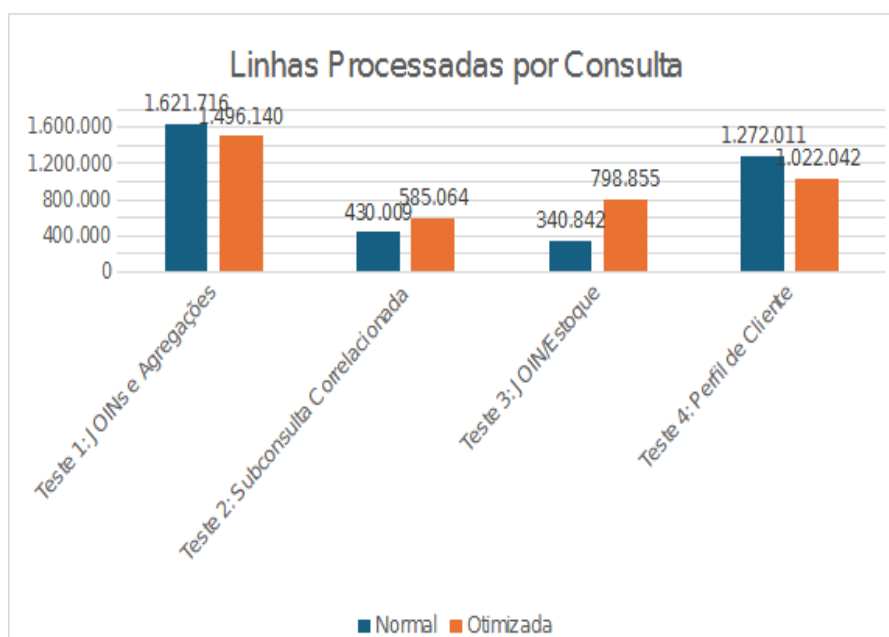


Figura 2 – Quantidade de Linhas Processadas por Consulta

Além da redução temporal, merece destaque a racionalização interna do SGBD provocada pelas reescritas via IA: há maior eficiência no uso de CTEs, minimização de operações repetitivas e aproveitamento de acessos mais inteligentes aos índices e tabelas. Trata-se de contribuições práticas que, na literatura (Oracle, 2025), já fundamentam bancos autônomos e SGBDs auto tunáveis em produção.

No entanto, o presente estudo também revela desafios importantes: certos tipos de consulta não apresentam sempre diminuição em linhas processadas, e a reescrita de queries exige conhecimento prévio sobre a estrutura e semântica dos dados para maximizar os benefícios da IA. O fenômeno se alinha à análise crítica de Marcus (2022), que aponta o risco de opacidade (caixa-preta) e de *overfitting* de modelos para determinados *workload*.

5 CONCLUSÃO

O objetivo central deste artigo – analisar, de forma sistemática e experimental, o impacto da utilização de técnicas de Inteligência Artificial para a otimização de consultas SQL – foi integralmente alcançado. Demonstrou-se, com dados objetivos e embasamento na literatura, que a aplicação de IA pode produzir melhorias expressivas no desempenho de bancos relacionais, promovendo redução do tempo de execução e maior eficiência no uso dos recursos computacionais em consultas complexas.

O atual estudo contribui com a consolidação do uso de ferramentas de IA no processo de reescrita e tuning de queries SQL, confirmando a relevância dessa abordagem perante workload desafiadores. A demonstração quantitativa, em ambiente controlado, do potencial de ganho de desempenho resultante dessas técnicas, com destaque para cenários de uso intensivo de subconsultas e múltiplos JOINS. O oferecimento de um roteiro metodológico replicável por pesquisadores e profissionais interessados em avaliar e aplicar IA na otimização

de bancos de dados, promovendo avanços na automação e auto ajuste desse tipo de sistema.

Como limitações, ressalta-se o uso de um banco de dados sintético e de ambiente controlado, que pode não contemplar toda a diversidade de situações em sistemas reais de larga escala e com cargas concorrentes. Recomenda-se, para trabalhos futuros, a aplicação da metodologia a bancos reais, distintos SGBDs e ambientes de nuvem, bem como a investigação do uso combinado de técnicas supervisionadas (machine learning tradicional) e não-supervisionadas (heurísticas evolutivas, reinforcement learning) para aprimorar ainda mais o processo de tuning automatizado.

Por fim, reitera-se que a IA, ainda que “fraca” em termos filosóficos, já se mostra capaz de agregar valor prático imediato a sistemas de banco de dados, estabelecendo uma nova fronteira para a gestão autônoma de informação.

REFERÊNCIAS

DOMINICO, L. B. **Tuning de Banco de Dados Relacional**: Conceitos, Técnicas e Ferramentas. Revista de Sistemas de Informação, p. 53-64, 2013.

ITCONVERGENCE. Oracle 23c AI Database: Self-Optimizing and Autonomous Features. Disponível em: <https://www.itconvergence.com/oracle-23c-ai-database/>. Acesso em: 10 ago. 2025.

MARCUS, R. et al. Neo: A Learned Query Optimizer. Proceedings of the VLDB Endowment, p. 1810-1822, 2022.

MENEZES, F. A.; SOUZA, L. M.; OLIVEIRA, T. R. **Otimização de Bancos de Dados com Inteligência Artificial**: Estado da Arte e Desafios. Journal of Database Technology, p. 23-45, 2023.

SELINGER, P. G. et al. Access Path Selection in a Relational Database Management System. Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data, p. 23-34, 1979.

SILVA, M. A. Aplicação de Técnicas de Inteligência Artificial para Tune Automatizado de SGBDs. **Anais do Simpósio Brasileiro de Banco de Dados**, p. 101-114, 2020.

SELINGER, P. Griffiths; ASTRAHAN, M. M. ; CHAMBERLIN, D. D. ; LORIE, R. A.; PRICE, T. G.. Access Path Selection: in a Relational Database Management System. **Duke University**, 1979. Disponível em: <https://courses.cs.duke.edu/compsci516/cps216/spring03/papers/selinger-et al-1979.pdf>. Acesso em: 28 jan. 2025.

MENEZES, Itala Vitória Coimbra Borges De; BARBOSA, Luiz Sergio De Oliveira Barbosa; GARZON, Newdon Ataíde Garzon. APLICAÇÕES DE INTELIGÊNCIA ARTIFICIAL: NA OTIMIZAÇÃO DE BANCO DE DADOS. **RECIMA21 REVISTA CIENTÍFICA MULTIDISCIPLINAR**, 2023. Disponível em: <https://recima21.com.br/recima21/article/view/4516/3166>. Acesso em: 30 jan. 2025.

MARCUS, Ryan. BAO: Making Learned Query Optimization Practical. **ACM Digital Library**, 2021. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/3448016.3452838>. Acesso em: 01 fev. 2025.

RUSSEL, Stuart; NORVIG, Peter. **Inteligencia Artificial**. Brasil: Elsevier Brasil, 2013. ISBN 978-85-352-5141-8.

MARCUS, Ryan; NEGI, Parimarjan; MAO, Hongzi; ZHANG, Chi; ALIZADEH, Mohammad; KRASKA, Tim; PAPAEMMANOUIL, Olga; TATBUL, Nesime. Neo: A Learned Query Optimizer. **ACM Digital Library**, 2019. Disponível em: <https://www.vldb.org/pvldb/vol12/p1705-marcus.pdf>, Acesso em: 02 mar. 2025.

DOMINICO, Simone. TUNING: UM ESTUDO SOBRE A OTIMIZAÇÃO DE DESEMPENHO DE SISTEMAS GERENCIADORES DE BANCO DE DADOS RELACIONAIS: SOB CARGA DE TRABALHO DE SUPORTE A DECISÃO. **UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**, 2013. Disponível em: https://repositorio.utfpr.edu.br/jspui/bitstream/1/11873/2/GP_COINT_2013_2_02.pdf. Acesso em: 02 mar. 2025.

MARCUS, Ryan; NEGI, Parimarjan; MAO, Hongzi; TATBUL, Nesime; ALIZADEH, Mohammad; KRASKA, Tim Kraska. Bao: Making Learned Query Optimization Practical. **ACM Digital Library**, 2021. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/3448016.3452838>. Acesso em: 03 mar. 2025.

2SQL, Ai. How AI is Transforming SQL Query Optimization in 2025. **AI2sql**, 2025. Disponível em: <https://ai2sql.io/how-ai-is-transforming-sql-query-optimization-2025>. Acesso em: 05 mar. 2025.

GLOBAL DEVELOPMENT GROUP, The PostgreSQL. Genetic Query Optimization (GEO) in PostgreSQL. **PostgreSQL Documentation**, 2025. Disponível em: <https://www.postgresql.org/docs/current/geqo-pg-intro.html>. Acesso em: 01 ago. 2025.

CALLAN, J; PETKE, J. Optimizing SQL Queries Using Genetic Improvement. **UCL Discovery**, 2021. Disponível em: <https://discovery.ucl.ac.uk/id/eprint/10122956/1/main.pdf>. Acesso em: 06 mar. 2025.

ORACLE, Oci Oracle. Autonomous Database Features. **Oracle**, 2025. Disponível em: <https://www.oracle.com/autonomous-database/features/>. Acesso em: 13 jul. 2025.

CONVERGENCE, It. Learn How Oracle Database 23ai Brings AI to your Data. **IT Convergence**, 2025. Disponível em: <https://www.itconvergence.com/blog/learn-how-oracle-database-23ai-brings-ai-to-your-data/>. Acesso em: 08 jun. 2025.