

### SQL como ele deve ser.

Guia de Boas Práticas em SQL Server para Desenvolvedores.

E aí, pessoal! Quem aí está pronto para otimizar seu código SQL e se tornar um ninja do SQL Server? Neste guia, vamos explorar algumas das melhores práticas para deixar suas consultas mais eficientes e seu banco de dados mais saudável. Vamos nessa!



Índices são como superpoderes para suas consultas! Eles melhoram a performance das consultas, mas é importante criá-los nas colunas certas. Priorize as colunas frequentemente usadas em cláusulas WHERE e JOIN. Mas cuidado para não criar índices em colunas que mudam pouco, senão seu índice pode virar um monstro gigante e pouco eficiente!

# USO DE ÍNDICES



**Indices** 

-- Criando um índice na tabela de clientes na coluna "nome" CREATE INDEX idx\_nome ON clientes (nome);



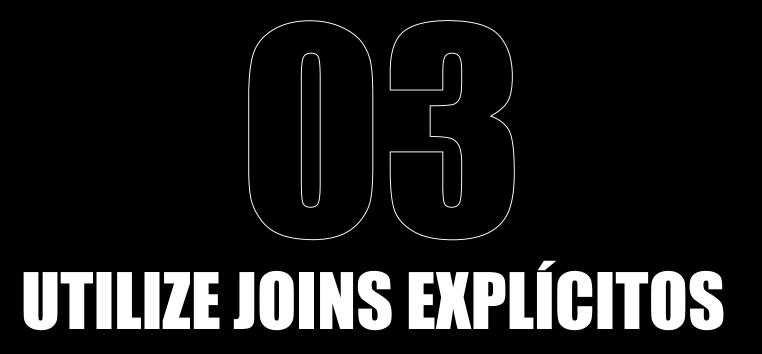
Ei, eu sei que é tentador, mas evite usar SELECT \*! Isso é como pedir um prato que você nem sabe o que vai ter dentro. Especifique apenas as colunas que você realmente precisa. Isso reduz a carga no servidor, melhora a legibilidade do código e ainda economiza nos recursos.

# EVITE USO DE (\*)



Uso do \*

-- Selecione apenas as colunas necessárias da tabela SELECT id, nome, email FROM clientes;



Já pensou em como seria confuso um encontro de amigos onde ninguém se conhece? Com o SQL, é a mesma coisa! Prefira JOINs explícitos (INNER JOIN, LEFT JOIN, etc.) em vez de JOINs implícitos. Isso deixa o código mais organizado e fácil de entender.

# UTILIZE JOINS EXPLÍCITOS



-- Utilizando INNER JOIN para combinar dados de tabelas SELECT c.nome, p.produto FROM clientes c INNER JOIN pedidos p ON c.id = p.cliente\_id;



# UTILIZE ALIASES DE TABELA

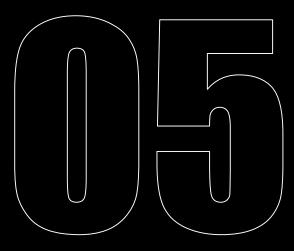
Sabe quando você chama seu amigo pelo apelido? Com as tabelas é a mesma coisa! Use aliases de tabela para simplificar suas consultas. Isso deixa o código mais conciso e fácil de entender, especialmente em consultas mais complexas.

### UTILIZE ALIASES DE TABELA



### **ALIASES**

-- Utilizando aliases de tabela para simplificar a consulta SELECT c.nome, p.produto FROM clientes AS c INNER JOIN pedidos AS p ON c.id = p.cliente\_id;



### EVITE FUNÇÕES EM COLUNAS DE FILTRO

Ah, as funções... Elas são como poções mágicas, mas às vezes podem atrapalhar! Evite usar funções em colunas de filtro (WHERE) para garantir que o SQL Server possa usar índices. Isso melhora o desempenho das consultas e evita dores de cabeça.

### EVITE FUNÇÕES EM COLUNAS DE FILTRO



### Funções

-- Evite usar funções em colunas de filtro
SELECT nome
FROM clientes
WHERE YEAR(data\_cadastro) = 2022;

Seguindo essas boas práticas, suas consultas serão mais eficientes e seu código SQL será muito mais fácil de entender e manter. Então bora lá, galera! Vamos nos tornar mestres do SQL Server e dominar o mundo das consultas eficientes!

### LinkedIn