**Verdades e mitos sobre JOINs**

19 de junho de 2012 por [Adeilson Brito](http://imasters.com.br/perfil/adeilson--brito/)

Durante os meus trabalhos de query tuning, é muito comum, por parte dos desenvolvedores, surgirem questões e mitos relacionados ao uso de JOIN. O tema pode denotar a falsa impressão de que é assunto para iniciantes, mas, eu garanto: não é! Faço uso do que Itzik Ben-Gan escreveu em seu livro “Inside Microsoft SQL Server 2008: T-SQL Querying”:

“O bom profissional é aquele que constrói o seu conhecimento a partir de fortes fundamentos, dominando o básico.”

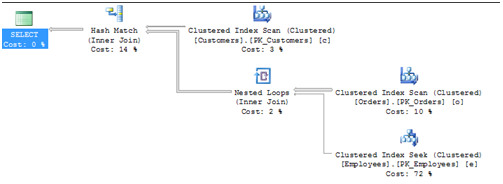
Sem mais delongas, vamos ao que interessa!

* Mito 1: Fazer JOIN na cláusula WHERE é menos performático do que na cláusula ON

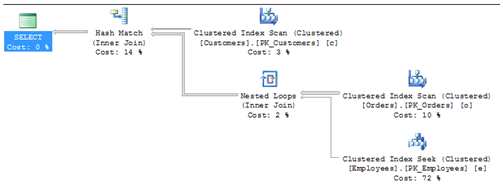
Inicialmente, quero destacar que fazer JOINs via WHERE é um estilo suportado pelo padrão ANSI SQL-89. Isso significa que JOIN via WHERE não é algo anti ANSI; pelo contrário: é um estilo de escrita devidamente padronizado. Já o uso de JOIN via cláusula ON foi formalmente regulado pelo ANSI SQL-92. Assim, ambos são válidos. A única diferença é que no WHERE não é possível implementar OUTER JOIN (somente INNER e CROSS). A Microsoft, contudo, criou uma abordagem proprietária, não reconhecida pela ANSI, visando fazer OUTER na cláusula WHERE: o uso do “\*=” e “=\*”. Vale lembrar que essa abordagem não é mais suportada a partir do SQL Server 2012, que tem nível mínimo de compatibilidade 90.

Superada essa questão, vamos, então, à próxima: existe diferença de performance? Vejamos os planos de execução de duas consultas, utilizando ambos os estilos:

use Northwind  
GO  
  
-- via cláusula ON  
SELECT o.OrderID, o.OrderDate, c.ContactName, e.LastName  
from dbo.Orders o  
join dbo.Customers c ON o.CustomerID = c.CustomerID  
JOIN dbo.Employees e ON o.EmployeeID = e.EmployeeID



-- via cláusula WHERE  
SELECT o.OrderID, o.OrderDate, c.ContactName, e.LastName  
from dbo.Orders o, dbo.Customers c, dbo.Employees e  
WHERE o.CustomerID = c.CustomerID  
and o.EmployeeID = e.EmployeeID



Como se vê, os planos são exatamente os mesmos. Não há benefício de performance ao usar um estilo ou outro. Particularmente, prefiro escrever como no primeiro exemplo, pois, para mim, a query fica com uma leitura mais intuitiva. Prefiro deixar os filtros para a cláusula WHERE e os joins para a cláusula ON.

* Mito 2: Quanto mais filtros na cláusula ON, mais rápida será a query

Essa é uma afirmação, às vezes, bastante defendida (equivocadamente). Primeiro, é importante compreender como o código de uma query é processado (não estou falando de geração de planos de execução). Segue o resumo da sequência lógica das fases de processamento do código fonte de uma query:

(5) SELECT  
(1) FROM [CROSS – INNER – OUTER]  
(2) WHERE  
(3) GROUP BY  
(4) HAVING  
(6) ORDER BY

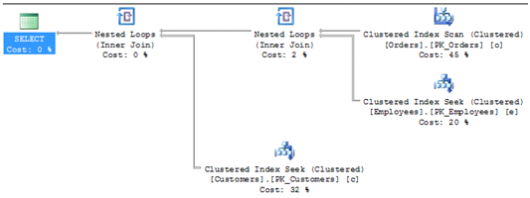
Note que, primeiramente, é processada a cláusula FROM e não o SELECT, que é a penúltima fase. A cláusula WHERE é a segunda a ser processada e assim sucessivamente.

Cada fase produz como resultado uma tabela virtual, que é utilizada como entrada na fase seguinte. A fase 1 – FROM pode possuir subfases quando a query envolver JOIN ou outros operadores de tabela, como PITVOT ou APPLY. Se tratando de JOINs, a sequência lógica das subfases do FROM é a seguinte:

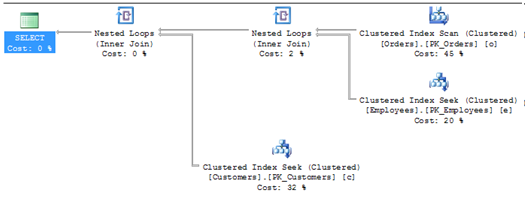
1. J1 – CROSS JOIN: é a primeira das três subfases. O CROSS JOIN gera um produto cartesiano combinando todas as linhas entre as duas tabelas envolvidas (N linhas x M linhas). O resultado desta subfase é uma tabela virtual contendo a combinação das linhas.
2. J2 – Filtro ON (INNER JOIN): esta subfase, se na query foi utilizada um INNER JOIN, recupera somente as linhas que atendam ao predicado da cláusula ON. O predicado ON é aplicado sobre todas as linhas retornadas na tabela virtual da subfase anterior (J1). O resultado da subfase J2 – Filtro ON é uma tabela virtual contendo exclusivamente as linhas que atendam ao predicado do ON.
3. J3 – OUTER JOIN: esta subfase, se na query foi utilizada um OUTER JOIN, adiciona à tabela virtual, produzida no passo anterior, as linhas da tabela preservada que não atenderam ao predicado do ON. Uma tabela é definida como preservada conforme o tipo de OUTER JOIN (Left, Right ou Full) utilizado. Por exemplo, num LEFT OUTER JOIN a tabela preservada é a da esquerda e, desta forma, as linhas que foram desconsideradas na subfase anterior, são agora adicionadas “de volta”, gerando como saída uma nova tabela virtual.

Com base no que acabamos de expor, depreendemos que o JOIN é processado antes do WHERE. Assim, vale destacar: não importa se o JOIN foi feito via cláusula ON ou se foi feito via cláusula WHERE, o SQL Server é “inteligente o bastante” para identificar a existência dos JOINs e processá-los antes dos filtros da query. Desta maneira, não faz diferença, para a performance da consulta, o local onde você escreve os filtros da sua query. Vamos clarificar as coisas através de um novo exemplo:

use Northwind  
GO  
  
-- INNER JOIN deixando os filtros na cláusula WHERE  
SELECT  
 o.OrderID, o.OrderDate, o.ShipRegion, o.ShipCountry,   
 c.CompanyName, c.Country, e.FirstName  
FROM orders o  
INNER JOIN dbo.Employees e ON o.EmployeeID = e.EmployeeID  
INNER JOIN dbo.Customers c ON o.CustomerID = c.CustomerID  
where o.ShipRegion IS NOT NULL  
AND o.ShipCountry = 'Brazil'



-- INNER JOIN aplicando os filtros diretamente na cláusula ON  
SELECT  
 o.OrderID, o.OrderDate, o.ShipRegion, o.ShipCountry,   
 c.CompanyName, c.Country, e.FirstName  
FROM orders o  
INNER JOIN dbo.Employees e  
 ON o.EmployeeID = e.EmployeeID  
INNER JOIN dbo.Customers c  
 ON o.CustomerID = c.CustomerID  
 AND o.ShipRegion IS NOT NULL  
 AND o.ShipCountry = 'Brazil'



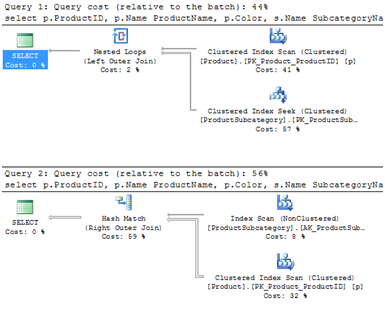
Veja que os planos de execução são os mesmos para ambas as consultas. Portanto, não há diferença de performance. Particularmente, prefiro escrever como no primeiro exemplo, pois visualmente fica mais fácil separar o que é predicado de JOIN e o que é predicado de WHERE.

Contudo, é necessário ter muito cuidado ao utilizar OUTER JOIN. Filtros adicionados ao predicado da cláusula ON de um OUTER JOIN podem prejudicar a performance, além de produzir um resultado errado.

Exploremos um exemplo:

use AdventureWorks  
go  
  
-- \*\*\* Listar os produtos da cor preta que tenham ou não subcategoria  
-- Mantém o filtro no WHERE  
select p.ProductID, p.Name ProductName, p.Color, s.Name SubcategoryName  
FROM Production.Product p  
LEFT OUTER JOIN Production.ProductSubcategory s  
 ON p.ProductSubcategoryID = s.ProductSubcategoryID  
WHERE p.Color = 'black'  
  
-- Traz o filtro para o ON do LEFT OUTER JOIN  
select p.ProductID, p.Name ProductName, p.Color, s.Name SubcategoryName  
FROM Production.Product p  
LEFT OUTER JOIN Production.ProductSubcategory s  
 ON p.ProductSubcategoryID = s.ProductSubcategoryID  
 and p.Color = 'black'

 Seguem os planos de execução de ambas as queries para comparação:



A query 1, com filtro no WHERE, retorna 93 linhas, enquanto que a query 2 retorna 504. Além do resultado divergente, a performance da segunda consulta é bem inferior à da primeira.

Neste ponto pergunto: qual query apresenta o resultado correto para o nosso exemplo? Ao trazer o filtro para a cláusula ON do LEFT OUTER JOIN na query 2, conforme já expliquei anteriormente, na subfase J2 – Filtro ON somente os produtos com cor preta são recuperados, mas, na subfase seguinte J3 – OUTER JOIN, as linhas que foram desconsideradas são adicionadas de volta (outer Rows). Assim, a segunda query não atende ao requisito especificado, pois lista os produtos da cor preta que tenham ou não subcategoria.

**Conclusão**

Não existe diferença de performance em fazer JOIN na cláusula WHERE ou na cláusula ON. No que diz respeito ao uso de filtros, também não há diferença de performance quando aplicados no WHERE ou totalmente no ON – faço aqui uma ressalva para os casos envolvendo OUTER JOIN.

**Webcast Otimizando Consultas T-SQL**

No dia 26 de junho, às 21 horas, estarei apresentando um webcast abordando o tema Otimizando Consultas T-SQL. Neste webcast irei apresentar maneiras de como analisar a performance das consultas e deixar dicas e melhores práticas.

Aos interessados, este é o [link](https://msevents.microsoft.com/CUI/EventDetail.aspx?EventID=1032515701&culture=pt-BR) para a inscrição.

**Mensagem do anunciante:**

Torne-se um Parceiro de Software Intel®. Filie-se ao Intel® Developer Zone. [Intel®Developer Zone](http://software.intel.com/pt-br/grow-business-reports?utm_source=IMasters&utm_medium=PaidMedia&utm_campaign=Partners-Brazil)

**Artigos relacionados**

[**Rastreamento de Exceções: Construindo um JAR executável – Parte 06**](http://imasters.com.br/linguagens/java/rastreamento-de-excecoes-construindo-um-jar-executavel-parte-06/)

[Desenvolvimento](http://imasters.com.br/secao/desenvolvimento/), [Java](http://imasters.com.br/secao/linguagens/java/) [Comente!](http://imasters.com.br/linguagens/java/rastreamento-de-excecoes-construindo-um-jar-executavel-parte-06/)

[**Programação Orientada a Objetos em 10 lições práticas – Parte 04**](http://imasters.com.br/desenvolvimento/visual-basic/programacao-orientada-objetos-em-10-licoes-praticas-parte-04/)

[.NET](http://imasters.com.br/secao/framework/dotnet/), [Desenvolvimento](http://imasters.com.br/secao/desenvolvimento/), [Visual Basic](http://imasters.com.br/secao/desenvolvimento/visual-basic/) [Comente!](http://imasters.com.br/desenvolvimento/visual-basic/programacao-orientada-objetos-em-10-licoes-praticas-parte-04/)

[**10 boas práticas para tornar seus aplicativos mais seguros – Parte 03**](http://imasters.com.br/infra/seguranca/10-boas-praticas-para-tornar-seus-aplicativos-mais-seguros-parte-03/)

[Desenvolvimento](http://imasters.com.br/secao/desenvolvimento/), [Segurança](http://imasters.com.br/secao/infra/seguranca/) [Comente!](http://imasters.com.br/infra/seguranca/10-boas-praticas-para-tornar-seus-aplicativos-mais-seguros-parte-03/)

[](http://imasters.com.br/perfil/adeilson--brito/)

**Adeilson Brito**

é graduado em Processamento de Dados, pós-graduado em Banco de Dados, especialista certificado em SQL Server (MCT, MCITP, MCTS, MCP). Experiência em administração e programação em banco de dados SQL Server, experiência em desenvolvimento de soluções BI usando a suíte SQL Server (SSAS / SSRS / SSIS), desenvolvimento de aplicações Web/Windows usando Dot Net C#.

[Página do autor](http://imasters.com.br/perfil/adeilson--brito/) [Email](mailto:adeilsonrbrito@gmail.com)

Leia os últimos artigos publicados por Adeilson Brito

* [Comparando semi joins](http://imasters.com.br/banco-de-dados/comparando-semi-joins/)
* [[SQL Server] Exame 70-461 – Tópico 4: Criando e Modificando Constraints](http://imasters.com.br/banco-de-dados/sql-server-exame-70-461-topico-4-criando-e-modificando-constraints/)
* [Melhores de 2012 – Verdades e mitos sobre JOINs](http://imasters.com.br/banco-de-dados/melhores-de-2012-verdades-e-mitos-sobre-joins/)
* [Melhores de 2012 – Consultando o Log de erro do SQL Server usando T-SQL](http://imasters.com.br/banco-de-dados/melhores-de-2012-consultando-o-log-de-erro-do-sql-server-usando-t-sql/)
* [Consultando o Log de erro do SQL Server usando T-SQL](http://imasters.com.br/banco-de-dados/consultando-o-log-de-erro-do-sql-server-usando-t-sql-2/)

[Comente também](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/#commentform)

**6 Comentários**

Fabiano Abreu

Adeilson,

Parabéns pela matéria, muito esclarecedora.

Abraços.

\_ \_  
Fabiano Abreu  
Papo Sql – Um blo com tutoriais, dicas e truques sobre SQL

[Há 2 anos](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/#comment-55207) [Responder](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/?replytocom=55207#respond)

Paulo

Gostei do artigo! DUVIDA!?

Então se tratando de query performace, se utilizar o FULL NAME da tabela gera uma melhor performace do que utilizando “Alias” ? e também em vez de usar “\*” digitar os nomes dos campos?? procede a informação se tratando de uma tabela com MUITOS dados?  
segue exemplo  
–Utilizando ALIAS  
SELECT t.\*,t2.campoX  
FROM tabela t  
LEFTJOIN tabela2 t2 ON t.campo1 = t2.campo1  
WHERE t.campo1 = XXXX  
/\*—————————————————————————————————–\*/  
–Utilizando FULL NAME TABELA  
SELECT tabela.campo1,tabela.campo2,tabela2.campoX  
FROM tabela  
LEFT JOIN tabela2 ON tabela.campo1 = tabela2.campo1  
WHERE tabela.campo1 = XXXX

[Há 2 anos](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/#comment-67623) [Responder](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/?replytocom=67623#respond)

Fabio da Silva

Muito bom o artigo. Até o recomendei no meu blog: <http://www.fabiophx.blogspot.com.br/2013/02/verdades-e-mitos-sobre-joins.html>

Ouvi tb q é aconselhável no where q a condição q irá filtrar mais fique por último e assim por diante.  
E tb q se tiver um índice com mesma ordem de campos utilizado no where esse será utilizado aumentando a performance.

Estão corretas?

Obrigado

[Há 2 anos](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/#comment-68535) [Responder](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/?replytocom=68535#respond)

Miquéias Rafael

Parabens mano pela publicação ajudou muito irmão, valew!!

[Há 1 ano](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/#comment-68751) [Responder](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/?replytocom=68751#respond)

José

No sentido de ampliar o que foi tratado no artigo a respeito do mito 1, “Fazer JOIN na cláusula WHERE é menos performático do que na cláusula ON”, existe uma possibilidade em que o join na cláusula ON pode oferecer melhor performance (no caso de inner join). Isto consta no livro “Inside Microsoft SQL Server 2008: T-SQL Querying”, anteriormente citado pelo autor do artigo:  
“… The one exception is when you specify GROUP BY ALL. Remember that GROUP BY ALL adds back groups that were filtered out by the WHERE clause, but it does not add back groups that were filtered out by the ON clause.”.  
O texto ressalta que “that this is a nonstandard legacy feature that you should avoid using.”.

Definir as ligações entre tabelas na cláusula JOIN, deixando os filtros para a cláusula WHERE, deixa o código-fonte mais legível, facilitando a compreensão e a manutenção.

[Há 1 ano](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/#comment-69269) [Responder](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/?replytocom=69269#respond)

**Qual a sua opinião?**

Parte superior do formulário



Parte inferior do formulário

[http://imasters.com.br/wp-content/uploads/2014/03/oferecimento_impacta1.png](http://www.impacta.com.br/)

**Cursos Relacionados**

**[Instalação e Gerência de lojas virtuais com Magento – Versão 1.4.0.1](http://cursos.imasters.com.br/instalacao-e-gerencia-de-lojas-virtuais-com-magento-versao-1-4-0-1?tracking=531dd2ffa6fd7" \o "Instalação e Gerência de lojas virtuais com Magento – Versão 1.4.0.1)**

**[Lógica de Programação – Primeiros passos](http://cursos.imasters.com.br/logica-de-programacao-primeiros-passos?tracking=531dd2ffa6fd7" \o "Lógica de Programação – Primeiros passos)**

[1](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/)

[2](http://imasters.com.br/artigo/24756/banco-de-dados/verdades-e-mitos-sobre-joins/)

**Últimas notícias**

15/08 às 04h35

[**Uber, AirBnB e outros serviços devem aceitar Bitcoin em breve**](http://imasters.com.br/noticia/uber-airbnb-e-outros-servicos-devem-aceitar-bitcoin-em-breve/)

15/08 às 04h00

[**Versão preview do Windows 9 pode chegar mês que vem**](http://imasters.com.br/noticia/versao-preview-windows-9-pode-chegar-mes-que-vem/)

15/08 às 03h40

[**Motorola deve revelar novas versões do Moto X e do Moto G em 4 de setembro**](http://imasters.com.br/noticia/motorola-deve-revelar-novas-versoes-moto-x-e-moto-g-em-4-de-setembro/)

15/08 às 02h30

[**Android sobe, enquanto iOS e Windows Phone caem no mercado de smartphones**](http://imasters.com.br/noticia/android-sobe-enquanto-ios-e-windows-phone-caem-mercado-de-smartphones/)

15/08 às 01h20

[**YouTube TV ganha update e novo visual**](http://imasters.com.br/noticia/youtube-tv-ganha-update-e-novo-visual/)

[Todas as Notícias »](http://imasters.com.br/noticias/)



**Twitter**

[Siga o perfil do iMasters](http://www.twitter.com/iMasters)



**LinkedIn**

[Cadastre-se no grupo iMasters](http://www.linkedin.com/company/imasters)



**gitHub**

[Códigos iMasters DEV](https://github.com/iMastersDev)



**RSS**

[Assine os feeds](http://imasters.com.br/feed/)

* [Impacta](http://www.impacta.com.br/)
* [DialHost](http://www.dialhost.com.br/?utm_source=parceiros_imasters&utm_medium=patrocinio&utm_campaign=Patrocinio%2BCanais%2BiMasters)
* [Magento](http://www.e-smart.com.br/)
* [SEBRAE](http://www.sebrae.com.br/)
* [Apiki](http://apiki.com/?utm_source=iMasters&utm_medium=Marca_Rodap%C3%A9&utm_campaign=Apiki)
* [Locaweb](http://www.locaweb.com.br/)
* [Caelum](http://www.caelum.com.br/)
* [EVEO](http://www.eveo.com.br/)
* [Intel](http://intel.ly/1es5qcS)
* [Site Blindado](http://www.siteblindado.com.br/)
* [Moip](http://www.moip.com.br/)
* [marcas_rodape_kinghost](http://www.kinghost.com.br/)

[iMasters Logo](http://imasters.com.br/)Copyright © 2014 Todos os direitos reservados

* [Sobre o iMasters](http://imasters.com.br/sobre-o-imasters/)
* [Política de Privacidade](http://imasters.com.br/politica-de-privacidade/)
* [Fale conosco](http://imasters.com.br/fale-conosco/)
* [iMasters Expert (english blog)](http://www.imasters.expert)

[O iMasters é hospedado por DialHost](http://www.dialhost.com.br/?utm_source=hospedado_por_imasters&utm_medium=patrocinio&utm_campaign=Patrocinio%2BCanais%2BiMasters)

**Social Media**

* [Twitter](http://www.twitter.com/iMasters)
* [Facebook](https://www.facebook.com/PortaliMasters)
* [LinkedIn](http://www.linkedin.com/company/imasters)
* [Flickr](http://www.flickr.com/photos/grupoimasters/)
* [RSS](http://imasters.com.br/feed/)

[faleconosco@imasters.com.br](mailto:faleconosco@imasters.com.br)

**Newsletter**

Fique por dentro de todas as novidades, eventos, cursos, conteúdos exclusivos e muito mais.

Parte superior do formulário





Parte inferior do formulário