

André Melancia
Microsoft Certified Trainer

SQL Indexing Tips

4 de maio de 2017

Agenda

- Introdução
- Tabelas
- Índices
- Armazenamento Físico
- Estatísticas
- Planos de Execução
- Conclusão



Introdução

Problemas a resolver:

- Lentidão geral
- Custos elevados de hardware e gestão da infraestruturas

Razão dos problemas:

- Mau desenho da base de dados
- Queries "abusivos"
- Concorrência de ligações
 - Excesso de ligações
 - Locks, deadlocks, etc.
- Planos de execução mal estimados
 - Más estatísticas
- Fragmentação
- Má configuração

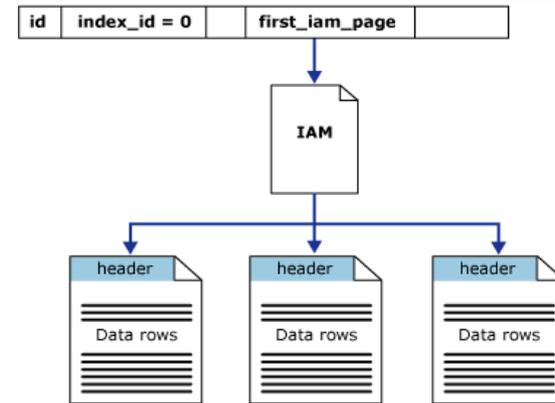
Armazenamento lógico:

- SQL Server guarda páginas de 8 KiB
 - Extend = 8 páginas = 64 KiB
- Cada página guarda 1 ou mais registos
 - Depende do tamanho dos registos
- Possibilidade de fragmentação
- Também se aplica a índices

Tabelas

Organização dos dados:

- Heap: Nenhuma ordem formal
 - Bom: inserção rápida de dados
 - Mau: pesquisa de dados
- Clustered: Os dados estão ordenados por um determinado critério
 - E.g. Chave Primária com ID numérico
 - Bom: se clustered por IDs sequenciais (INT ou BIGINT)
 - Mau: se GUIDs, valores aleatórios (datas nascimento, localidade, etc.)



O que é um índice?

- Definido numa tabela
- Regista fisicamente uma "sub-tabela" de dados
 - Apenas 1 ou mais campos
 - Em alguns casos, apenas alguns registos
- Na execução de queries:
 - Pode substituir a leitura numa tabela
 - Um índice é JOINed de forma semelhante a uma tabela
- Implementado com B-Tree
- Uma Chave Primária (PK) é um índice

Pesquisas em tabelas e índices:

- Scan: Lê os registos do início ao fim da tabela/índice
 - $O(N/2)$
 - 1000k registos precisam em média de ler metade das páginas
- Seek: Procura os registos através da B-Tree do índice
 - $O(\text{Log}(N))$
 - 1000k registos precisam de ler pouco mais de 5 páginas

Utilidade?

- Desempenho
 - Leitura, mas também escrita (INSERT, UPDATE e DELETE)
- Limitador
 - Chave Primária (garante unicidade nos dados, sem NULLs)
 - Index Unique (garante unicidade nos dados, NULLs permitidos opcionalmente)

Muitos ou poucos?

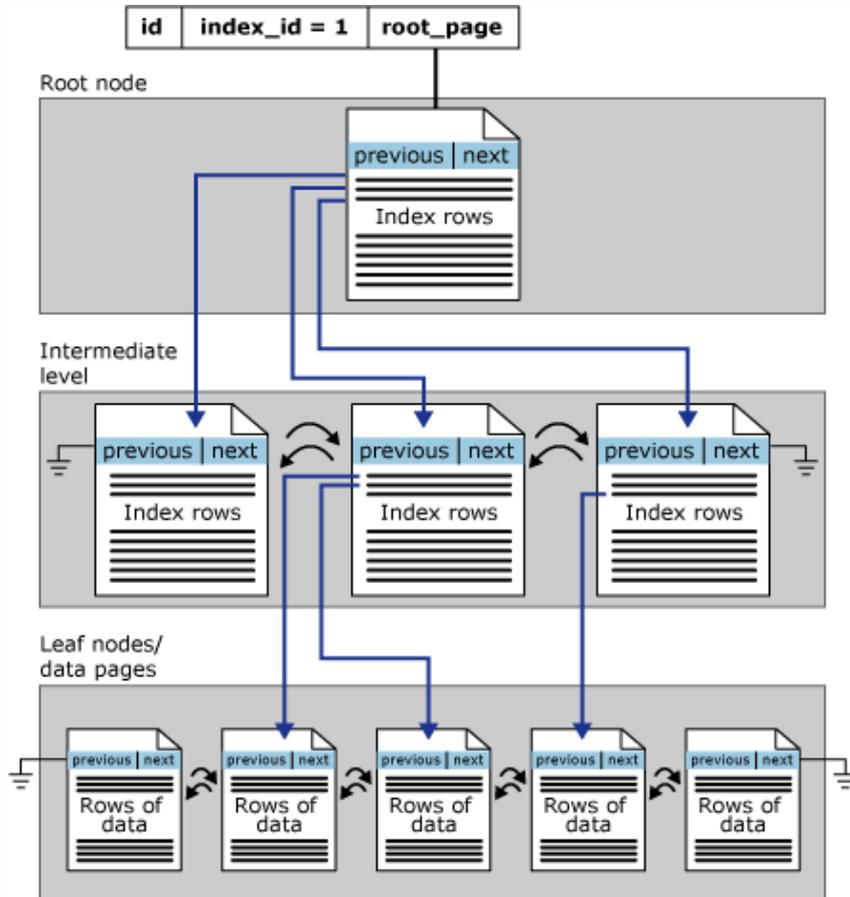
- Ausência de índices:
 - Necessidade de scans (lentos), escritas rápidas
- Excesso de índices:
 - Leituras rápidas, demasiado espaço ocupado, demora em escritas

Tipos de índices:

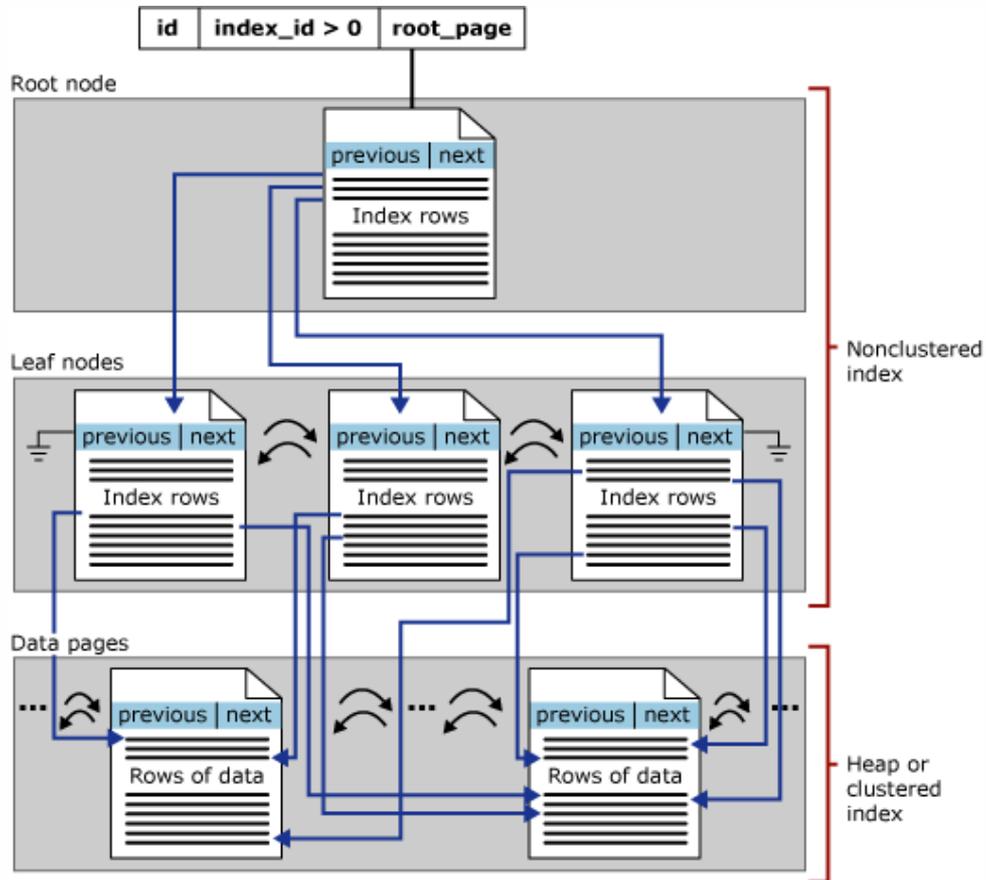
- Clustered
- Non-Clustered
- Columnstore
- Full-Text
- Spatial
- XML

Índices

Clustered

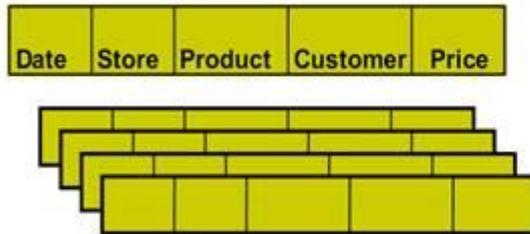


Non-Clustered

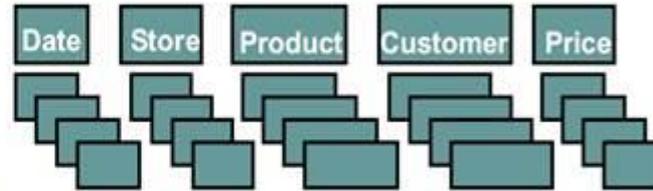


Columnstore

row-store



column-store



Outras características opcionais (1):

- **Unique (por omissão é Non-Unique)**
 - `CREATE UNIQUE INDEX UX_Pessoas_NIF ON Pessoas (NIF);`
- **Covering indexes**
 - `CREATE NONCLUSTERED INDEX IX_Pessoas_Info ON Pessoas (ID_Pessoa ASC, Nome_Pessoa ASC) INCLUDE (Localidade, EMail);`
 - Não precisa de consultar a tabela original. Como o índice ocupa menos páginas, a leitura é mais rápida

Outras características opcionais (2):

- Partitioned indexes
 - Criados numa ou mais partições numa tabela particionada
- Filtered indexes
 - Limita os registos referenciados pelo índice
 - ```
CREATE NONCLUSTERED INDEX IX_Pessoas_Info ON Pessoas (ID_Pessoa ASC, Nome_Pessoa ASC) WHERE Activo = 1;
```

## Configurações de índices:

- Fill factor:
  - Reserva espaço "livre" para inserção de futuros registos (e assim evita fragmentação)
- Pad Index:
  - Nós intermédios da B-Tree também utilizam fill factor
- Recomendações:
  - Bom: manter o valor recomendado por omissão
  - Mau: definir um valor muito alto (aumenta fragmentação ao inserir novos registos) ou um valor muito baixo (demasiadas leituras de storage)

## Bulk loading:

- Em processos nocturnos regulares (ETL), etc.
- Se uma parte significativa da tabela for apagada/alterada, pode ser mais rápido remover os índices e recriá-los no final do processo
  - Difícil de estimar - medir as várias opções

## Problemas típicos (1):

- Ausência ou excesso de índices
- Mau fill factor e fragmentação
- Repetição de índices
  - Repetições puras:  $IX(A, B)$  duas vezes
  - Repetições equivalentes:  $IX(A \text{ ASC}, B \text{ ASC}) = IX(A \text{ DESC}, B \text{ DESC})$
- Sub-conjuntos:
  - $IX(A, B)$  e  $IX(A)$

## Problemas típicos (2):

- Má ordem:
  - `IX (B, A)` em vez `IX (A, B)`
- Ordenação
  - `IX (A ASC, B ASC)` quando poderia ser mais útil `IX (A ASC, B DESC)`
- Campos irrelevantes na hierarquia
  - `IX (A, B)` em vez `IX (B)` quando só é preciso B

# Armazenamento físico

## Onde?

- Discos locais
- SAN
- Windows Storage Spaces
- Azure Blob storage

# Armazenamento físico

## Tipos de discos:

- Mecânicos
- SSD
- Outros:
  - Híbridos
  - Virtuais
  - Remotos

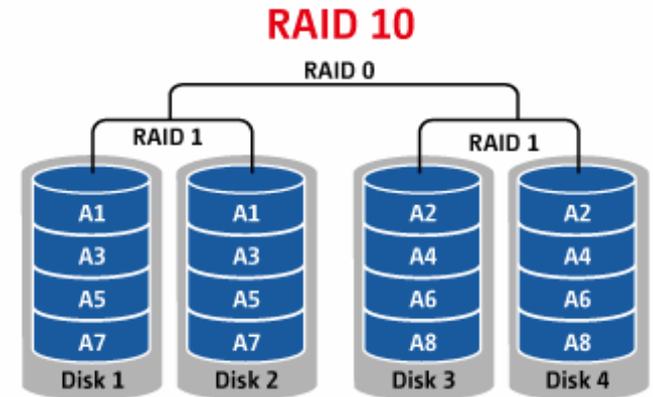
## RAID:

- Níveis de RAID

- 0 (striping)
- 1 (mirror)
- 10 = 1 + 0
- 5 (múltiplos discos com 1 disco de redundância)
- 6 (múltiplos discos com 2 disco de redundância)

- Recomendação:

- Transaction Logs: RAID 10
- Dados (populares): RAID 10
- Dados (normais): RAID 5 ou 6



## Recomendações:

- Evitar particionar os discos (usar discos físicos diferentes)
- Alinhar partições a fronteiras físicas dos discos
- Formatar com unidade de alocação (cluster size) a 64 KiB
  - Recomendado para SSDE e SSAS
- Com striping RAID 0 ou 10 usar também stripes de 64 KiB se possível

## Onde armazenar?

- Múltiplos discos
  - OS, Pagefile, TempDB, bases de dados (separar dados e log), etc.
- Opção: índices e tabela respectiva em discos diferentes
- Opção: particionamento das tabelas
  - Índices podem aplicar-se apenas a uma ou mais partições
- Opção: compressão de dados
  - Mais lento a descomprimir em CPU mas compensa a poupança no tempo de leitura no disco

## Filegroups

- Evitar múltiplos filegroups no mesmo disco físico
- Filegroups diferentes quando dados que são acedidos em conjunto
  - E.g. Tabelas (ou índices) que são JOINed regularmente

Porquê?

- Para poder estimar planos de execução

Como?

- Informação geral sobre número de registos, valores em cada campo, etc.

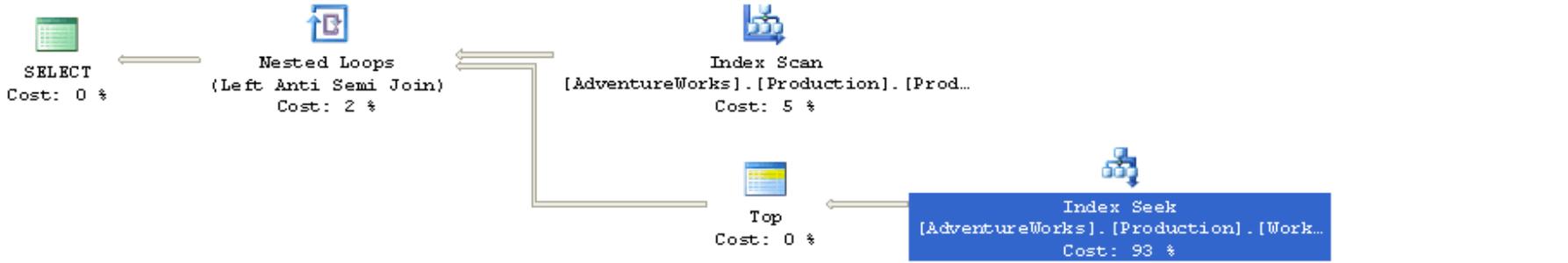
- Ver estatísticas:
  - DBCC SHOW\_STATISTICS (tabela, índice) WITH HISTOGRAM;
- Configuração automática:
  - ALTER DATABASE base\_de\_dados SET  
AUTO\_CREATE\_STATISTICS { ON | OFF }  
AUTO\_UPDATE\_STATISTICS { ON | OFF }  
AUTO\_UPDATE\_STATISTICS\_ASYNC { ON | OFF }

- Forçar a actualização de estatísticas
  - Quando há uma percentagem grande de alterações
  - Quando os valores estimados e reais dos planos de execução são muito diferentes
  - UPDATE STATISTICS tabela;
  - UPDATE STATISTICS tabela índice;
  - EXEC SP\_UpdateStats;

# Planos de execução

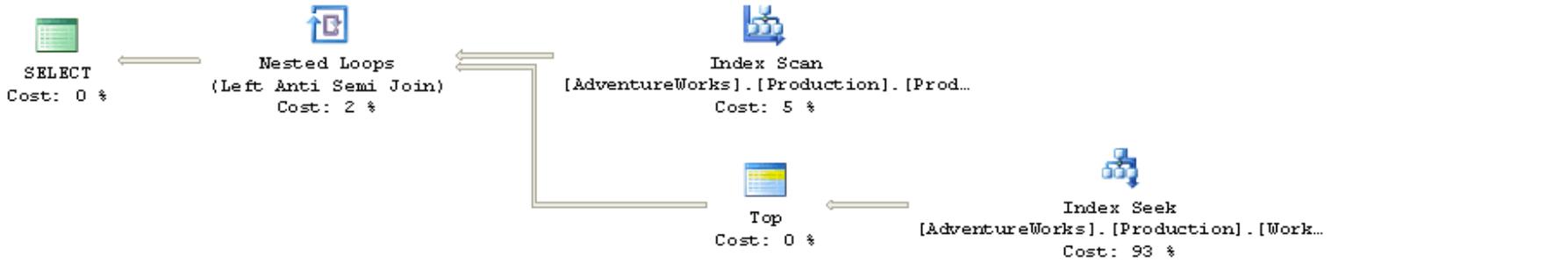
Results Messages Execution plan

Query 1: Query cost (relative to the batch): 50%  
SELECT ProductID FROM Production.Product EXCEPT SELECT ProductID FROM Production.WorkOrder ;



The execution plan for Query 1 shows a 'SELECT' operation with a cost of 0% on the left. It is connected to a 'Nested Loops (Left Anti Semi Join)' operation with a cost of 2%. This operation is connected to two other operations: an 'Index Scan [AdventureWorks].[Production].[Prod...]' with a cost of 5% and a 'Top' operation with a cost of 0%. The 'Top' operation is connected to an 'Index Seek [AdventureWorks].[Production].[Work...]' with a cost of 93%, which is highlighted in blue.

Query 2: Query cost (relative to the batch): 50%  
SELECT ProductID FROM Production.Product WHERE ProductID NOT IN ( SELECT ProductID FROM Production.WorkOrder);



The execution plan for Query 2 is identical to Query 1. It shows a 'SELECT' operation with a cost of 0% on the left, connected to a 'Nested Loops (Left Anti Semi Join)' operation with a cost of 2%. This operation is connected to an 'Index Scan [AdventureWorks].[Production].[Prod...]' with a cost of 5% and a 'Top' operation with a cost of 0%. The 'Top' operation is connected to an 'Index Seek [AdventureWorks].[Production].[Work...]' with a cost of 93%.

Query executed successfully.

- Database Engine Tuning Advisor
- DMVs
- Query Store
- Profiler
- [...]
  
- Sentry One
  - SQL Sentry
  - Plan Explorer
- Red Gate
  - SQL Index Manager

# Questões?

*Why does  
a round pizza  
come in a  
square box?*



<http://www.actualtraining.pt/P.aspx?T=2&id=M10987>

[Fabricantes](#) > [Microsoft](#) > [Database](#) > [SQL Server](#) > [SQL Server 2016](#) > M10987

## > Performance Tuning and Optimizing SQL Databases

**Duração do Curso:** 4 Dia(s) - 24:00 Horas

**Código do Curso:** M10987

**Preço:** 1090,00€

 Pedido de Informações

 Imprimir

 Conteúdo programático

 Partilhar

Mostrar tudo / Esconder tudo

### ↑ Introdução

This four-day instructor-led course provides students who manage and maintain SQL Server databases with the knowledge and skills to performance tune and optimize their databases.

### ↓ Público-Alvo

### ↑ Quando completar o curso

After completing this course, students will be able to:

- Describe the high level architectural overview of SQL Server and its various components.
- Describe the SQL Server execution model, waits and queues.
- Describe core I/O concepts, Storage Area Networks and performance testing.
- Describe architectural concepts and best practices related to data files for user databases and TempDB.
- Describe architectural concepts and best practices related to Concurrency, Transactions, Isolation Levels and Locking.
- Describe architectural concepts of the Optimizer and how to identify and fix query plan issues.
- Describe architectural concepts, troubleshooting scenarios and best practices related to Plan Cache.
- Describe architectural concepts, troubleshooting strategy and usage scenarios for Extended Events.
- Explain data collection strategy and techniques to analyze collected data

### Datas Agendadas

**Inscryva-se  
Maio**

29/Mai - 01/Jun  
(D2)

**Julho**

10 - 13 (D2)

**Agosto**

21 - 24 (D2)

**Informação de horários:**  
D2 • Dias úteis das 09:30 às  
17:00 (6h)

## André Melancia

Microsoft Certified Trainer

<http://Andy.PT>

Actual Training – Consultoria e Formação, Lda.  
Rua Cupertino de Miranda, 6 - 2B  
Edifício Multitech  
1600-546 Lisboa  
Portugal



# Obrigado!

