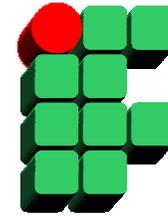


Formação de DBAs SQL Server 2008

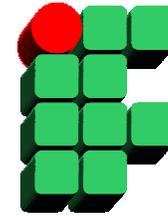
Parte 6: Gerenciamento de Performance

Monitoramento



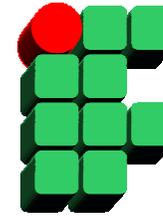
- O monitoramento é primeira etapa do gerenciamento de performance. Monitorar significa estabelecer uma linha de base, ou seja, registrar os dados referentes ao ambiente para servirem de base para decisões pró-ativas e reativas relacionadas à performance;
- Diversos fatores podem influenciar a performance de um SGBD como CPU, memória, I/O, bloqueios e deadlocks. Monitorar esses elementos é fundamental para garantir uma performance adequada à carga do sistema.

Ferramentas de Monitoramento



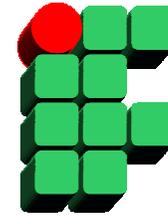
- System Monitor (Performance Monitor)
 - Oferece um conjunto de contadores sobre o sistema operacional e do SQL Server 2008 que podem ser utilizados para definir um baseline monitorar o sistema;
 - Principais contadores:
 - *Memory: Available MBytes*: Memória disponível em megabytes;
 - *Paging File: %Usage*: Percentual de utilização do arquivo de paginação;
 - *Physical Disk: Avg. Disk Queue Length*: Valor médio da fila do subsistema de I/O;
 - *Processor: %Processor Time*: Percentual de utilização de CPU;
 - *Process: %Processor Time*: Percentual de utilização de CPU de um processo específico;
 - *Network Interface: Bytes Total/sec*: Total de bytes enviados e recebidos pela interface de rede.

Ferramentas de Monitoramento



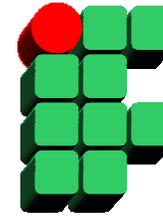
- System Monitor (Performance Monitor)
 - Principais contadores:
 - *SQLServer:Access Methods - Range Scans/sec*: Quantidade de range scans, em índices, por segundo;
 - *Full Scans/sec*: Quantidade de full scans por segundo;
 - *Index Searches/sec*: Quantidade de buscas, através de índices, por segundo;
 - *Worktables Created/sec*: Quantidade de worktables criadas por segundo;
 - *SQLServer:SQL Statistics - Batch Requests/sec*: Número de batches recebidas por segundo;
 - *SQL Compilations/sec*: Número de compilações SQL por segundo;
 - *SQL Re-Compilations/sec*: Número de recompilações SQL por segundo.

Ferramentas de Monitoramento



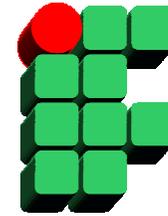
- System Monitor (Performance Monitor)
 - Principais contadores:
 - *Database Transaction/sec*: Número de transações por segundo;
 - *Transactions - Longest Transaction Running Time*: Duração da maior transação em atividade no banco de dados em questão;
 - *Locks – Average Wait Time (ms)*: Tempo médio de espera por recursos bloqueados;
 - *Lock Wait Time (ms)*: Tempo total de espera por recursos bloqueados;
 - *Lock Requests/sec*: Quantidade de locks por segundo;
 - *Lock Waits/sec*: Quantidade de requisições que resultaram em espera por segundo.

Ferramentas de Monitoramento



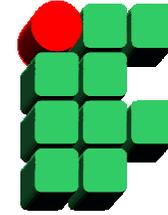
- SQL Profiler
 - Usado para monitorar eventos de uma instância do SQL Server 2008;
 - Principais contadores:
 - *RPC:Completed*: Chamadas RPC concluídas;
 - *SP:Completed*: Stored procedures concluídas;
 - *SP:StmtCompleted*: Comandos individuais concluídos em uma stored procedure;
 - *SQL:StmtCompleted*: Comandos SQL concluídos;
 - *SQL:BatchCompleted*: Batches SQL concluídas;
 - *Lock:Acquired*: Indica que a transação monitorada conseguiu um lock sobre um recurso;
 - *Lock:Released*: Indica que a transação monitorada liberou um lock sobre um recurso;
 - *Lock:Timeout*: Indica que uma requisição de lock foi cancelada por timeout em virtude de um bloqueio.

Otimização



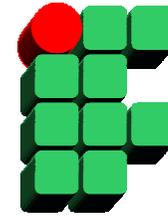
- Um sistema otimizado tende a consumir menos recursos e atingir uma performance maior do que um sistema não otimizado. Pode-se dizer então que otimizar é ajustar configurações, estruturas de dados e códigos de aplicação de forma a obter uma performance adequada, consumindo o mínimo de recursos;
- Fatores importantes no processo de otimização de um sistema de banco de dados:
 - Monitoramento;
 - Configurações do SGBD;
 - Organização física de cada banco de dados;
 - Boas práticas de programação;
 - Análise de planos de execução;
 - Criação de índices adequados.

Trabalhando com índices



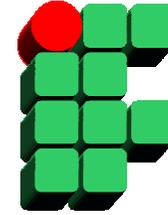
- Como os dados são armazenados:
 - Os registros são armazenados em páginas de dados;
 - Pilhas são coleções de páginas de dados de uma tabela.
- Como os dados são acessados:
 - Examinando todas as páginas de dados das tabelas;
 - Usando um índice que aponte para dados em uma página.

Criar ou não criar índices?



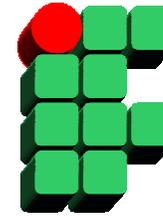
- Por que criar índices?
 - Índices aceleram a recuperação dos dados pois reduzem a quantidade de operações de I/O necessárias para localizar os registros;
 - Podem ser usados para garantir a unicidade dos registros.
- Por que evitar a criação de um índice?
 - Índices consomem espaço em disco adicional;
 - Operações de atualização ficam sobrecarregadas em virtude da necessidade de atualizar os índices.

Índices Nonclustered



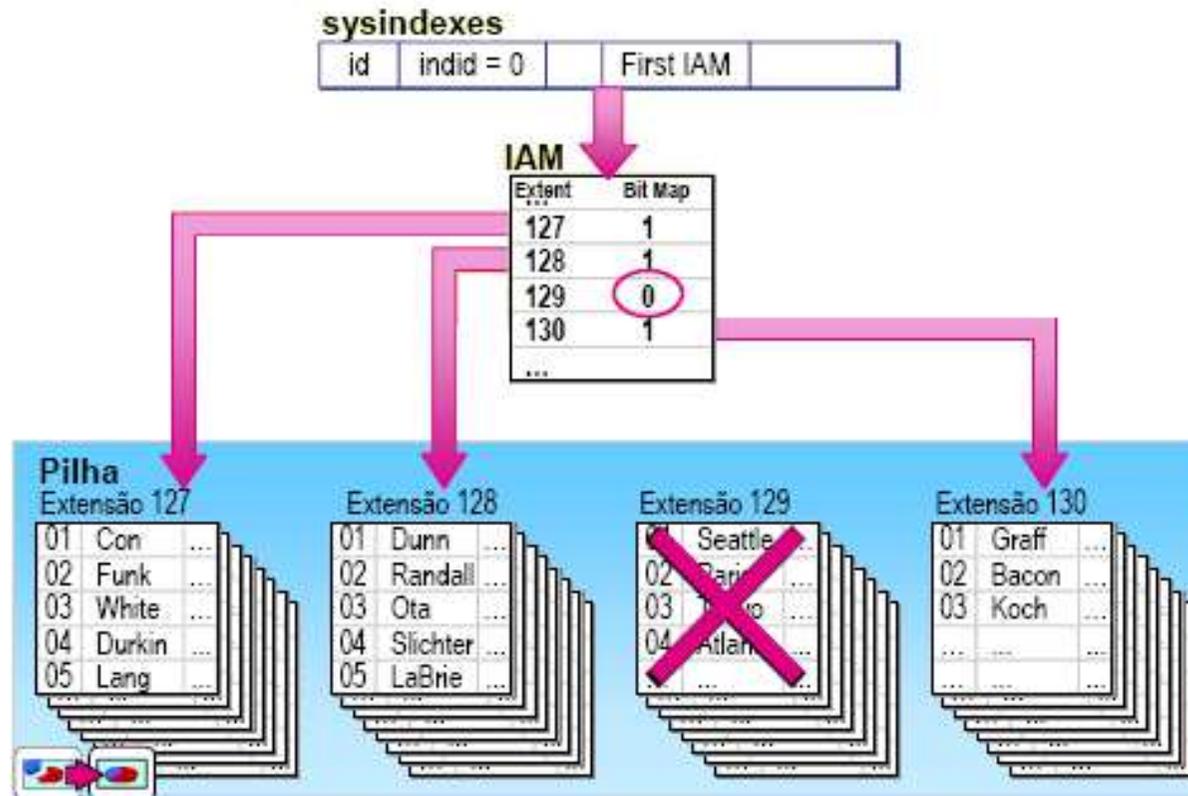
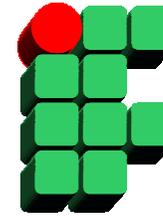
- Não exigem ordenação física dos dados;
- Opção padrão para índices secundários;
- Leaf pages são ponteiros para os dados;
- Menor overhead durante atualizações.
- São recriados na seguintes circunstâncias:
 - Se um índice clustered for criado;
 - Quando um índice clustered existente é excluído.

Índices Clustered

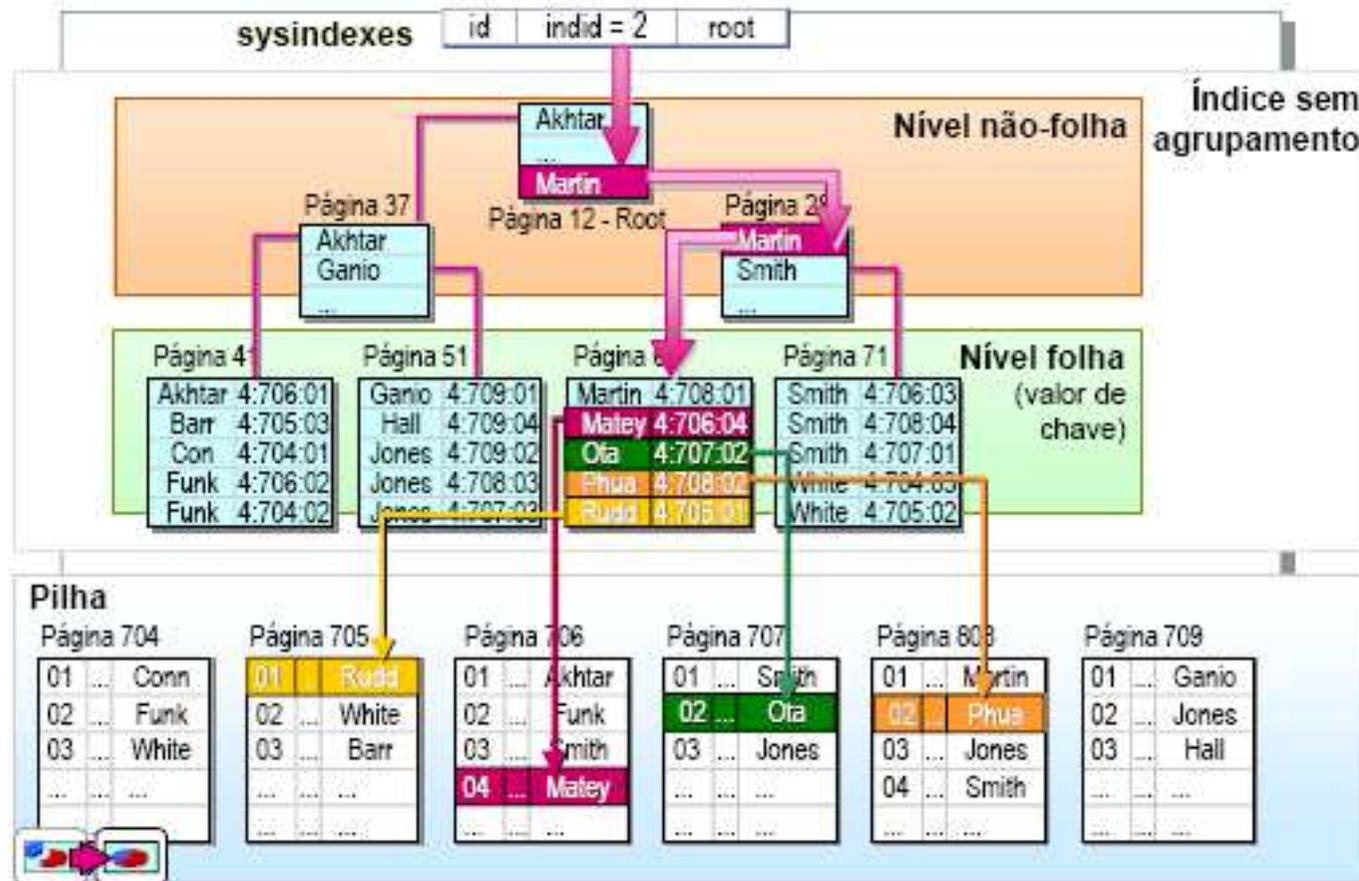
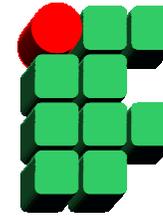


- Forçam uma ordenação física dos dados;
- Opção padrão para chaves primárias;
- Leaf pages são as próprias páginas de dados;
- Maior overhead durante atualizações;
- Cada tabela pode ter apenas um índice clustered.

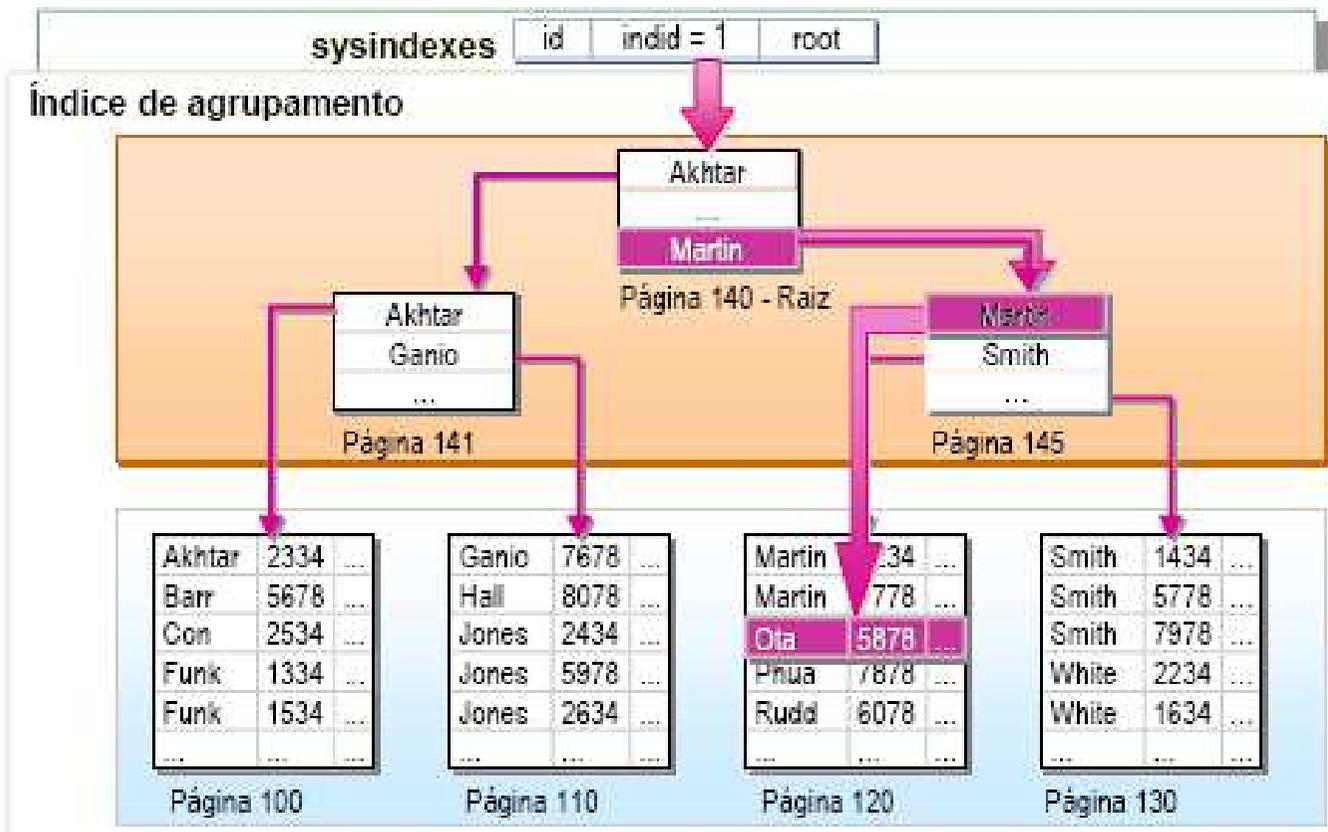
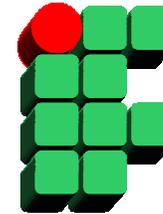
Localizando registros sem índice



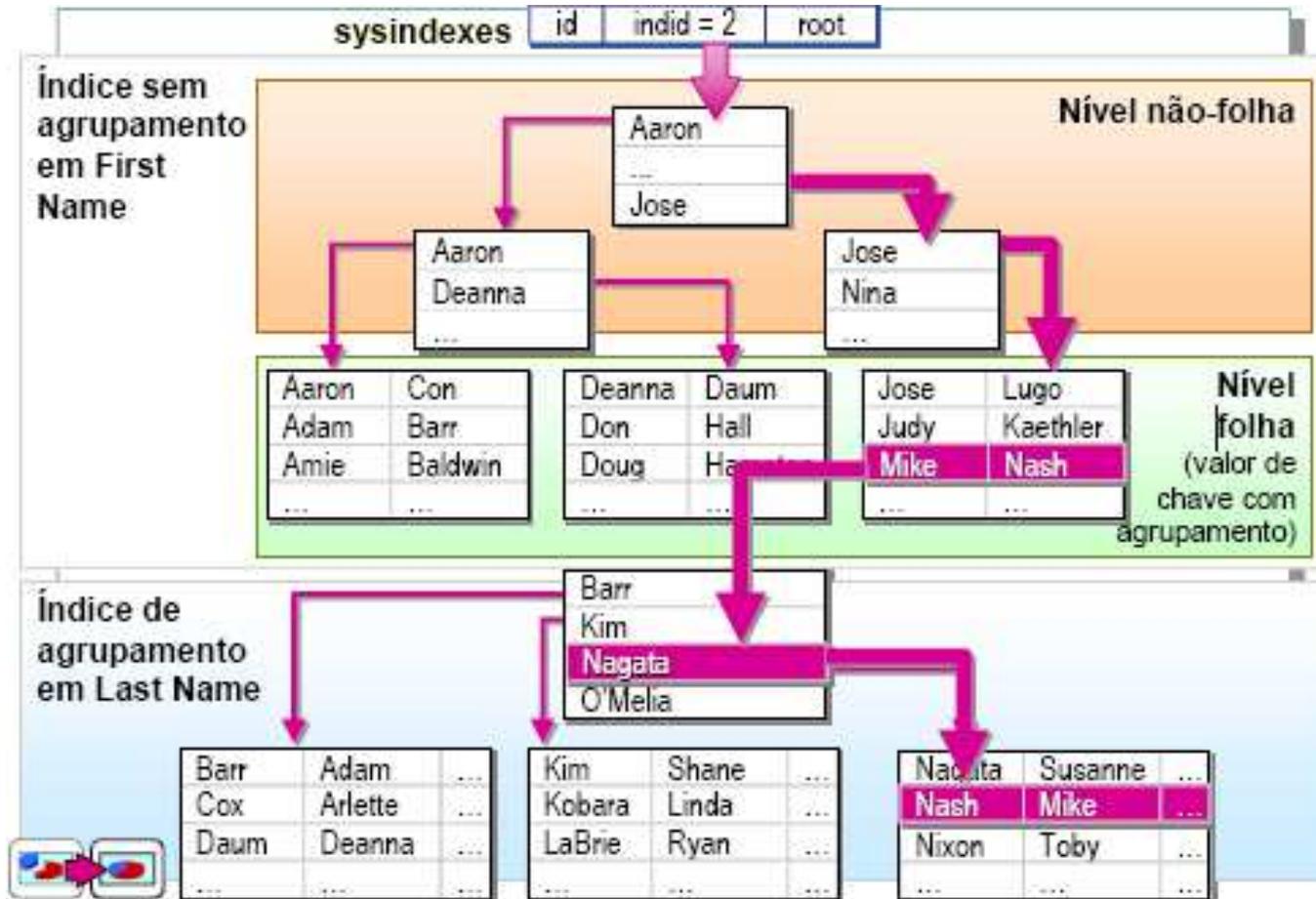
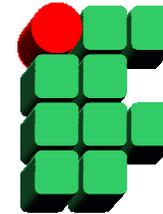
Localizando registros com índice nonclustered



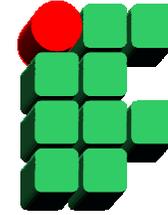
Localizando registros con índice clustered



Localizando registros com índice nonclustered sobre um índice clustered

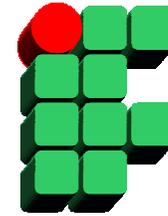


Decidindo que colunas indexar



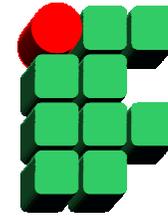
- Conhecer os dados a serem indexados:
 - A estrutura física e lógica;
 - As características dos dados;
 - Como os dados são usados:
 - Os tipos de consulta utilizados;
 - A frequência das consultas normalmente realizadas.

Diretrizes sobre Indexação



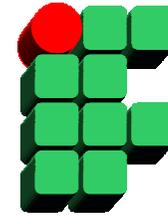
- Colunas que devem ser indexadas:
 - Chaves primárias e estrangeiras;
 - Colunas pesquisadas para a localização por faixas;
 - Colunas acessadas para consultas ordenadas;
 - Colunas agrupadas durante agregações.
- Colunas que não devem ser indexadas:
 - Colunas raramente referenciadas em uma consulta;
 - Colunas que contenham poucos valores únicos (baixa seletividade).

Diretrizes sobre Indexação



- Quando usar índices clustered:
 - Tabelas que sofrem pouca atualização;
 - Consultas por intervalo (> , < , between) ou consultas que exijam ordenação pelos campos do índice;
 - Consultas em tabelas que possuam um grande número de registros com valores distintos nos campos do índice;
 - Consultas que retornem grandes volumes de dados;
 - Índices clustered devem possuir um número reduzido de colunas, com poucos caracteres e tipos de dados que ocupem pouco espaço em disco.

Diretrizes sobre Indexação

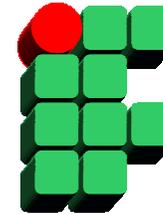


- Quando usar índices nonclustered:
 - Tabelas que sofrem atualização frequentemente;
 - Índices que possuam muitos campos;

A Bad Clustered Index Is Worse Than No Index

“I have occasionally run across good nonclustered indexes that point to a very bad clustered index. For example, suppose that we have a table that represents everybody in the country. Furthermore, let’s say that we create a clustered index for the respective states in which everybody lives. In addition, we create nonclustered indexes on last name, first name, and state. Upon performing an index lookup on last name, first name, and state, the nonclustered index, the bookmark key is returned in only a few page selects. In this example, the very fast nonclustered.”

Determinando a seletividade



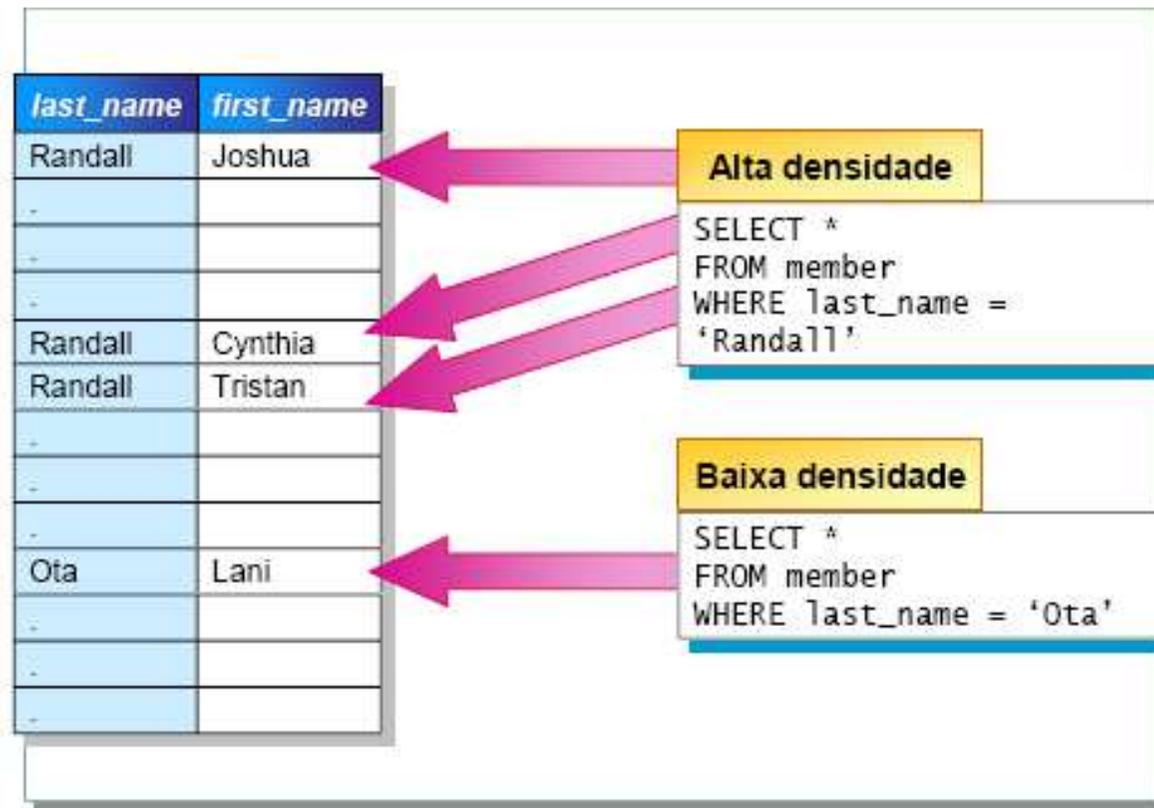
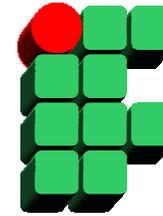
<i>member_no</i>	<i>last_name</i>	<i>first_name</i>	
1	Randall	Joshua	Alta seletividade Número de registros que atendem aos critérios = $\frac{1000}{10000} = 10\%$ Número total de registros em uma tabela
2	Flood	Kathie	
.			
.			
.			
10000	Anderson	Bill	

```
SELECT *  
FROM member  
WHERE member_no > 9001
```

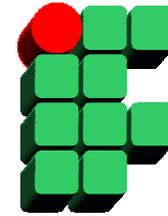
<i>member_no</i>	<i>last_name</i>	<i>first_name</i>	
1	Randall	Joshua	Baixa seletividade Número de registros que atendem aos critérios = $\frac{9000}{10000} = 90\%$ Número total de registros em uma tabela
2	Flood	Kathie	
.			
.			
.			
10000	Anderson	Bill	

```
SELECT *  
FROM member  
WHERE member_no < 9001
```

Determinando a densidade

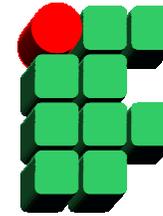


Planos de Execução



- Otimizador: módulo do SQL Server responsável por definir a melhor estratégia para a execução dos comandos Transact-SQL;
- Plano de Execução: estrutura de dados que armazena a definição de um ou mais comandos Transact-SQL. É a principal estrutura utilizada pelo Otimizador para cumprir os seguintes objetivos:
 - Minimizar a compilação de comandos;
 - Selecionar índices apropriados para cada comando;
 - Definir se a consulta deve ser paralelizada.

Planos de Execução

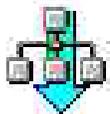


- Interpretação:

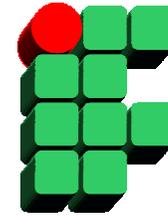
- Table Scan – Varredura em toda a tabela para buscar as informações desejadas. Possui um alto custo de I/O e prejudica a concorrência. É uma operação eficiente quando a diferença entre quantidade de registros desejada e a quantidade total de registros da tabela é pequena;



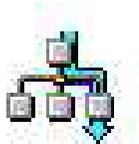
- Index Scan – Varredura completa em um índice. Possui um custo menor de I/O, mas ainda assim prejudica a concorrência. É uma operação eficiente para consultas por intervalo ($>$, $<$, between) ou para agregações em campos presentes no índice.



Planos de Execução



- Interpretação:

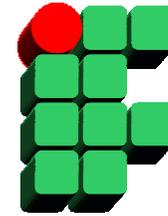


- Index Seek – Busca em um conjunto restrito de registros para encontrar as informações desejadas. Possui um baixo custo de I/O e é a forma ideal para maximizar a concorrência. É uma operação eficiente quando as cláusulas de filtro (ON, WHERE, HAVING) são bastante restritivas;



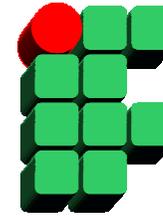
- Bookmark Lookup – A partir das chaves (RID) encontradas no índice, busca os registros correspondentes na tabela. Quando todos os campos desejados estão presentes no índice, essa operação é dispensada.

Estatísticas



- Estatísticas são Informações sobre os dados de uma coluna em uma tabela (frequência e distribuição dos valores), usadas pelo otimizador para estimar o custo de uma consulta e selecionar o(s) índice(s) adequado(s);
- A criação de um índice gera automaticamente estatísticas sobre as colunas daquele índice;
- Podem ser criadas estatísticas para colunas não indexadas;
- Estatísticas precisam ser atualizadas regularmente para garantir a eficiência do otimizador;
- A criação e a atualização das estatísticas pode ser feita de forma automática ou manual.

Trabalhando com estatísticas

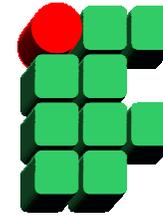


- Criando Estatísticas:

```
CREATE STATISTICS statistics_name
  ON { table | view } ( column [ ,...n ] )
  [ WITH
    [ [ FULLSCAN
      | SAMPLE number { PERCENT | ROWS } ][ , ] ]
    [ NORECOMPUTE ]
  ]
```

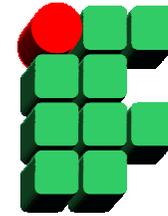
FULLSCAN	Faz uma varredura completa na tabela para gerar as estatísticas
SAMPLE	Determina uma amostra baseada em um percentual (PERCENT) ou na quantidade de registros (ROWS) a ser examinada
NORECOMPUTE	Impede que as estatísticas sejam atualizadas automaticamente pelo SQL Server

Trabalhando com estatísticas



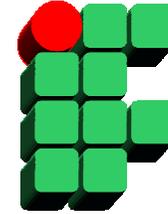
- Atualizando Estatísticas:
 - Para uma tabela ou visão: UPDATE STATISTICS
 - Para todas as tabelas de um banco de dados: SP_UPDATESTATS
- Visualizando Estatísticas:
 - DBCC SHOW_STATISTICS
- Automatizando a criação e atualização de estatísticas:
 - ALTER DATABASE nome_banco SET AUTO_CREATE_STATISTICS ON
 - ALTER DATABASE nome_banco SET AUTO_UPDATE_STATISTICS ON

Fragmentação



- Como ocorre a fragmentação:
 - O SQL Server modifica páginas de índice quando os dados são modificados;
 - A reorganização faz com que a estrutura do índice fique fragmentada.
- Avaliando a fragmentação de um índice:
 - A visão de gerenciamento `sys.dm_db_index_physical_stats` indica o grau de fragmentação de um índice.

Fragmentação

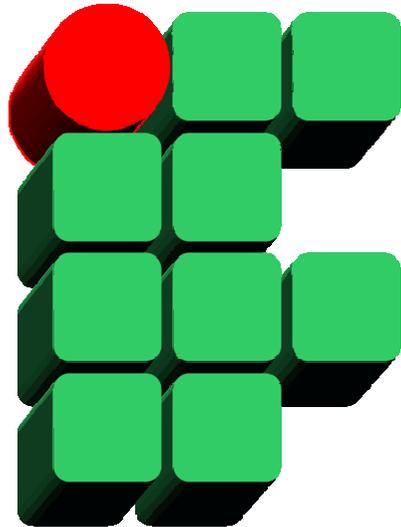


- Desfragmentando índices:
 - Se a fragmentação de um índice for inferior a 30%, é recomendada a execução do comando ALTER INDEX REORGANIZE. Esse comando reordena o nível folha (leaf pages) do índice;

```
ALTER INDEX idx_employee ON HumanResources.Employee REORGANIZE
```

- Se a fragmentação de um índice for superior a 30%, é recomendada a execução do comando ALTER INDEX REBUILD. Esse comando recria toda a estrutura do índice. Por ser uma operação custosa, o rebuild deve ser executado em conjunto com a cláusula ONLINE.

```
ALTER INDEX idx_employee ON HumanResources.Employee  
REBUILD WITH ONLINE = ON
```



Formação de DBAs SQL Server 2008

Parte 6: Gerenciamento de Performance