

BANCO DE DADOS II

Gerenciamento de armazenamento



Equipe

Charlyane Araújo
Hugo Júnior
Matheus Nascimento
José Leão

CONTEÚDO



01

PÁGINAS



02

EXTENDS



03

ÍNDICES



04

LINKED SERVER

PÁGINAS

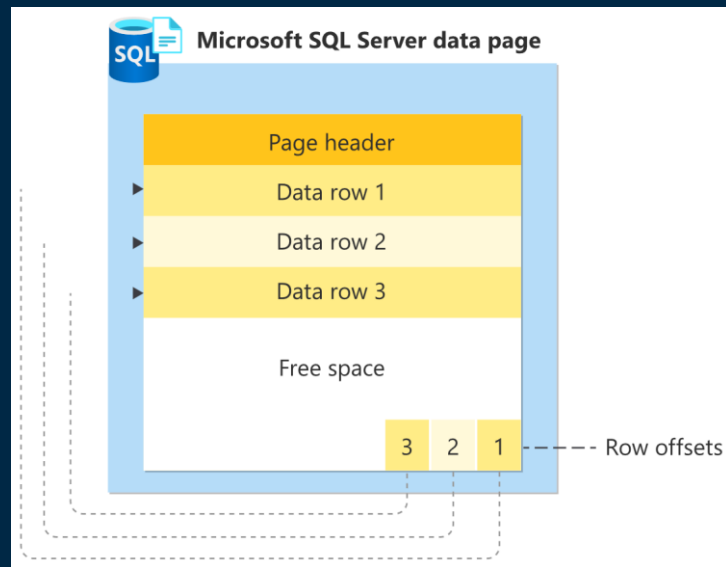
01

PÁGINAS

A unidade fundamental de armazenamento de dados no SQL Server é a página. O espaço em disco alocado a um arquivo de dados (.mdf ou .ndf) em um banco de dados é logicamente dividido em páginas numeradas de forma contígua de 0 a n.

Todas as páginas de dados têm o mesmo tamanho: 8kb. Cada página começa com um cabeçalho de 96 bytes, onde são armazenadas as seguintes informações: número de página, tipo de página, quantidade de espaço livre na página e a ID de unidade de alocação do objeto que possui a página.

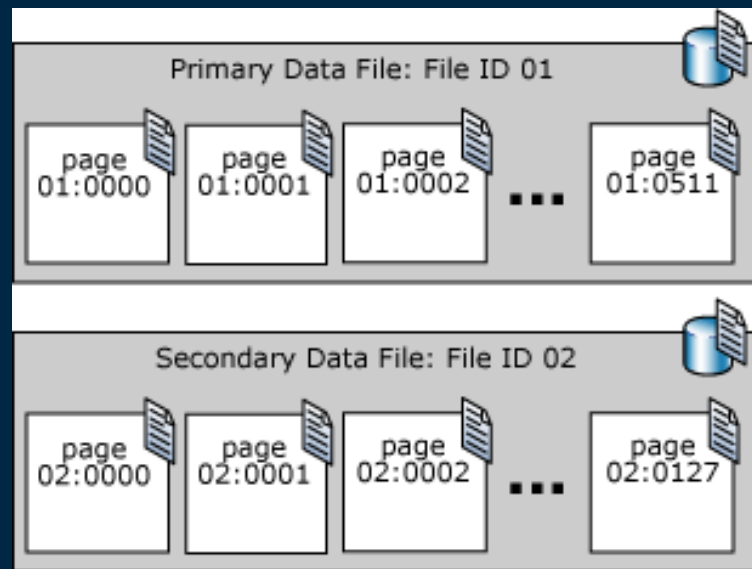
As linhas de dados são armazenadas em série na página, iniciando logo após o cabeçalho. Uma tabela de compensação da linha tem início no final da página, e cada tabela de compensação da linha contém uma entrada para cada linha na página. Cada entrada de compensação de linha armazena a distância do primeiro byte da linha em relação ao início da página, sendo a sua principal função ajudar o SQL Server a localizar linhas em uma página rapidamente. Essas entradas estão em sequência inversa da sequência das linhas na página.



PÁGINAS

As páginas de um arquivo de dados do SQL Server são numeradas em sequência, iniciando com zero (0) na primeira página do arquivo. Cada arquivo em um banco de dados tem um número de ID de arquivo exclusivo. Para identificar de forma exclusiva uma página em um banco de dados, são necessários ID do arquivo e número de página. O exemplo a seguir mostra os números de página em um banco de dados que tem um arquivo de dados primário de 4 MB e um arquivo de dados secundário de 1 MB.

Uma página de cabeçalho de arquivo é a primeira página que contém informações sobre os atributos do arquivo. Várias outras páginas do início do arquivo também têm informações de sistema, como mapas de alocação. Uma das páginas de sistema armazenada no arquivo de dados primário e no primeiro arquivo de log é uma página de inicialização de banco de dados que contém informações sobre os atributos do banco de dados.



PÁGINAS

TIPO DE PÁGINA

SUMÁRIO

Dados

Linhas de dados com todos os dados, exceto os dados de texto, ntext, imagem, nvarchar(max), varchar(max), varbinary(max) e xml, quando o texto na linha estiver definido como ON.

Índice

Entradas de índice.

Texto/Imagem

Tipos de dados de objeto grande: dados text, ntext, image, nvarchar(max), varchar(max), varbinary(max) e xml.

Colunas de comprimento variável quando a linha de dados exceder 8 KB: varchar, nvarchar, varbinary e sql_variant.

GAM (Global Allocation Map) SGAM (Shared Global Allocation Map)

Informações sobre alocação de extensões.

PÁGINAS

TIPO DE PÁGINA

SUMÁRIO

PFS (Espaço Livre na Página)

Informações sobre alocação de página e espaço livre disponível em páginas.

IAM (mapa de alocação de índice)

Informações sobre extensões usadas por uma tabela ou índice por unidade de alocação.

BCM (Bulk Changed Map)

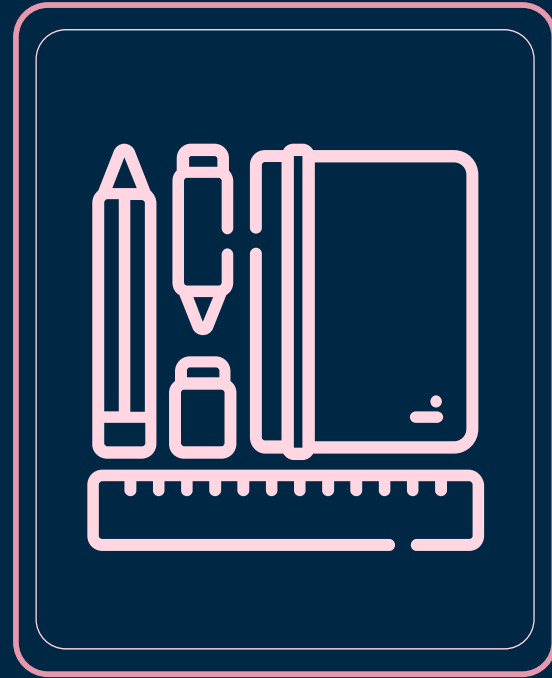
Informações sobre extensões modificadas pelas operações em massa desde a última instrução BACKUP LOG por unidade de alocação.

DCM (Differential Changed Map)

Informações sobre extensões modificadas desde a última instrução BACKUP DATABASE por unidade de alocação.

PÁGINAS

Exemplo Prático



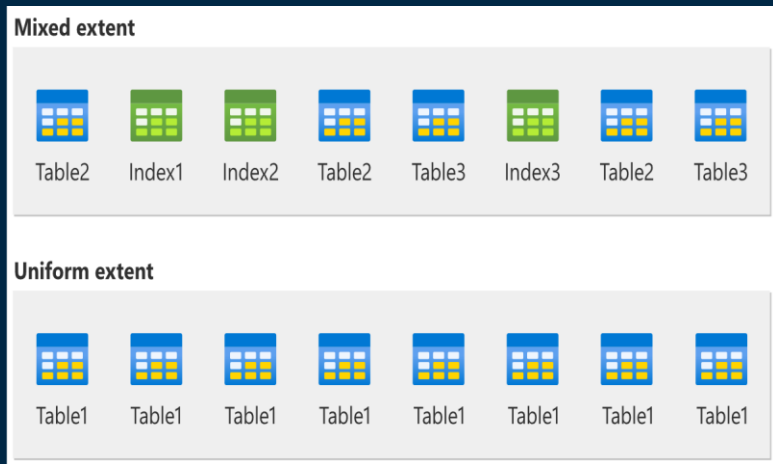
EXTENDS

02

EXTENDS

- Extends são a unidade básica em que o espaço é gerenciado. Uma extensão tem oito páginas fisicamente contíguas ou 64 KB. Isso significa que os bancos de dados do SQL Server têm 16 extensões por megabyte.

- O SQL Server tem dois tipos de extends:
 - Extensões **uniformes** que pertencem a um único objeto; todas as oito páginas na extensão podem ser usadas apenas pelo objeto proprietário.
 - Extensões **mistas** compartilhadas por até oito objetos. Cada uma das oito páginas da extensão pode pertencer a um objeto diferente.



EXTENDS

Gerenciar alocações de extensões e espaço livre

- As estruturas de dados do SQL Server que gerenciam alocações de extensão e controlam espaço livre têm uma estrutura relativamente simples. Ele oferece as seguintes vantagens:
- As informações de espaço livre são compactadas, portanto, poucas páginas contêm essas informações. Isso aumenta a velocidade reduzindo o número de leituras de disco exigidas para recuperar as informações de alocação.
- A maioria das informações de alocação não é encadeada. Isso simplifica a manutenção das informações de alocação.

Gerir Alocações

- O SQL Server usa dois tipos de mapas de alocação para registrar a alocação de extensões:
- GAM(Global Allocation map):
 - As páginas GAM registram quais extensões foram alocadas. Cada GAM cobre 64.000 extensões ou quase 4GB de dados. O GAM tem 1 bit para cada extensão no intervalo coberto.

SGAM(Shared Global Allocation Map):

- As páginas SGAM registram quais extensões estão sendo usadas atualmente como extensões mistas e também têm pelo menos uma página não usada

EXTENDS

- Cada extensão tem os padrões de bit a seguir configurados no GAM e no SGAM, com base em seu uso atual

Uso atual da extensão	Configuração de bit GAM	Configuração de bit SGAM
Livre, não está sendo usado	1	0
Extensão uniforme ou extensão mista completa	0	0
Extensão mista com páginas livres	0	1

EXTENDS

Acompanhar o espaço livre

As páginas **PFS (Page Free Space)** registram o status de alocação de cada página, se uma página individual foi alocada e a quantidade de espaço livre em cada página. O PFS tem 1 byte para cada página, que registra se a página está alocada e, em caso afirmativo, se ela está vazia, de 1 a 50% completa, de 51 a 80% completa, de 81 a 95% completa ou de 96 a 100% completa.

Após a alocação de uma extensão a um objeto, o Mecanismo de Banco de Dados usa as páginas PFS para registrar quais páginas na extensão são alocadas ou livres. Essas informações são usadas quando o Mecanismo de Banco de Dados precisa alocar uma página nova. A quantidade de espaço livre em uma página é mantida apenas para páginas heap e de texto/imagem. Ela é usada quando o Mecanismo de Banco de Dados precisa encontrar uma página com espaço livre disponível para manter uma linha recentemente inserida. Como o ponto em que a linha nova deve ser inserida é definido pelos valores de chave de índice, os índices não exigem que o espaço livre da página seja controlado.

Uma nova página PFS, GAM ou SGAM é adicionada ao arquivo de dados para cada intervalo adicional que ele acompanha. Portanto, há uma nova página PFS localizada a 8.088 páginas após a primeira página PFS e páginas PFS adicionais em intervalos de 8.088 páginas seguintes. Para ilustrar isso, a ID da página 1 é uma página PFS, a ID da página 8088 é uma página PFS, a ID da página 16176 é uma página PFS etc.

Há uma nova página GAM localizada a 64.000 extensões após a primeira página GAM e ela acompanha os intervalos de 64.000 extensões posteriores; a sequência continua em intervalos de 64.000 extensões. Da mesma forma, há uma nova página SGAM localizada a 64.000 extensões após a primeira página SGAM e páginas SGAM adicionais nos intervalos de 64.000 extensões seguintes.

EXTENDS

Gerenciar o espaço usado por objetos

- Uma página **IAM** mapeia as extensões em uma parte de 4 GB de um arquivo de banco de dados usada por uma unidade de alocação. Uma unidade de alocação deve ser de um dos três tipos:
 - **IN_ROW_DATA** -> Mantém uma partição de um heap ou um índice.
- **LOB_DATA** -> Contém tipos de dados de LOB (objeto grande), como **xml**, **varbinary(max)** e **varchar(max)**.
 - **ROW_OVERFLOW_DATA** -> Mantém dados de comprimento variável armazenados em colunas **varchar**, **nvarchar**, **varbinary** ou **sql_variant** que excedem o limite de tamanho de linha de 8.060 bytes

EXTENDS

Exemplo Prático



ÍNDICES

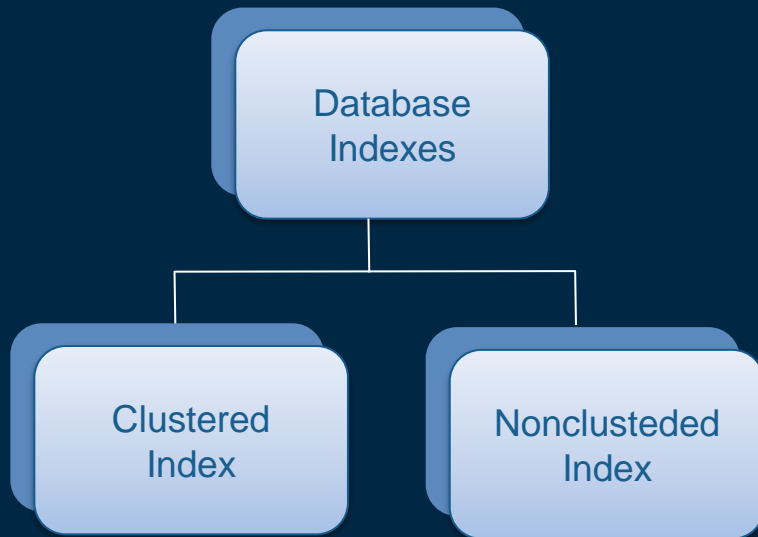
03

ÍNDICES

São estruturas de acesso auxiliares, utilizadas para agilizar a recuperação de registros em resposta a certas condições de pesquisa.

As estruturas de índice, são arquivos adicionais no disco que oferecem *caminhos de acesso secundários*, os quais permitem formas alternativas de acessar os registros sem afetar seu posicionamento físico no arquivo de dados primário no disco.

Os índices são uma das ferramentas de otimização mais conhecidas e utilizadas pelos desenvolvedores de Banco de dados.



Índice Clustered

- Índices clusterizado alteram a forma como os dados são armazenados em banco de dados, pois ele classifica as linhas de acordo com a coluna que possui o índice.
- Ele é montado na própria tabela, criando a estrutura ORDENADA de árvore para facilitar as buscas.
- Uma tabela só pode ter um índice clusterizado.
- Cada folha do índice clusterizado possui todas as informações do registro.
- Se uma tabela não possuir índice clusterizado, suas linhas são armazenadas em uma estrutura não ordenada chamada de HEAP.
- Exemplo clássico: Dicionário (Você localiza a palavra e junto com ela já tem a definição).

-- Criando tabela ESTUDANTE

```
CREATE TABLE ESTUDANTE (  
Codigo_Estudante INT,  
Nome_Estudante VARCHAR(50),  
Idade_Estudante INT,  
Genero_Estudante VARCHAR(1),  
Curso_Estudante VARCHAR(50) );
```

-- Inserindo registros na tabela ESTUDANTE

```
INSERT INTO ESTUDANTE (Codigo_Estudante, Nome_Estudante, Idade_Estudante, Genero_Estudante, Curso_Estudante)  
VALUES (1, 'José Alberto', 20, 'M', 'Administração de Empresas'),  
(2, 'Maurício Lima', 19, 'M', 'Matemática'),  
(3, 'Davi Leão', 25, 'M', 'Ciência da Computação'),  
(4, 'Virginia Batista', 30, 'F', 'Marketing'),  
(5, 'Amélia Cristina', 22, 'F', 'Geografia'),  
(6, 'Simone Leticia', 24, 'F', 'Letras');
```

-- Criando índice clusterizado na coluna Codigo_Estudante

```
CREATE CLUSTERED INDEX PK_Codigo_Estudante  
ON ESTUDANTE(Codigo_Estudante);
```

ÍNDICE Nonclustered

- Em um índice não clusterizado a forma como os dados são armazenados não é alterada, e um objeto separado é criado na tabela, apontando para as linhas da tabela original após a busca.
- É uma estrutura ORDENADA à parte, que contém apenas a coluna indexada (e as colunas do INCLUDE, caso exista) e uma tabela pode ter N índices deste tipo.
- Se for necessário consultar alguma informação que não está no índice NONCLUSTERED, a informação é localizada utilizando o índice clusterizado da tabela (Key Lookup)
- Exemplo clássico: Índice de um livro (Você localiza no índice a página onde está o capítulo e depois vai até a página para ver as informações).

```
-- Criando tabela ESTUDANTE
```

```
CREATE TABLE ESTUDANTE (  
Codigo_Estudante INT,  
Nome_Estudante VARCHAR(50),  
Idade_Estudante INT,  
Genero_Estudante VARCHAR(1),  
Curso_Estudante VARCHAR(50) );
```

```
-- Inserindo registros na tabela ESTUDANTE
```

```
INSERT INTO ESTUDANTE (Codigo_Estudante, Nome_Estudante, Idade_Estudante, Genero_Estudante, Curso_Estudante)  
VALUES (1, 'José Alberto', 20, 'M', 'Administração de Empresas'),  
(2, 'Mauricio Lima', 19, 'M', 'Matemática'),  
(3, 'Davi Leão', 25, 'M', 'Ciência da Computação'),  
(4, 'Virginia Batista', 30, 'F', 'Marketing'),  
(5, 'Amélia Cristina', 22, 'F', 'Geografia'),  
(6, 'Simone Letícia', 24, 'F', 'Letras');
```

```
-- Criando índice não clusterizado na coluna Curso_Estudante
```

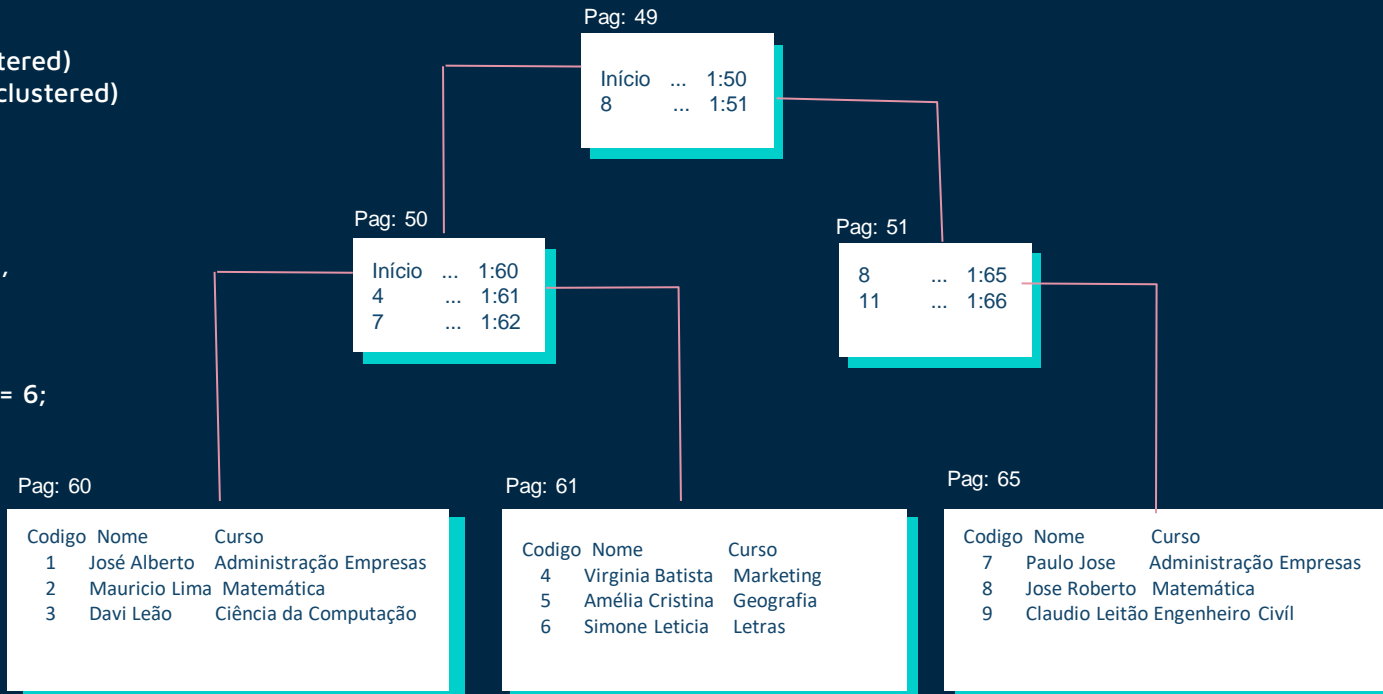
```
CREATE NONCLUSTERED INDEX IDX_Curso_Estudante  
ON ESTUDANTE(Curso_Estudante);
```

Cenário: Índice Clustered

TABELA com 5 colunas:

- `Codigo_Estudante` (Clustered)
- `Curso_Estudante` (Nonclustered)
- `Nome_Estudante`
- `Idade_Estudante`
- `Genero_Estudante`

```
Select Codigo_Estudante,  
       Nome_Estudante,  
       Curso_Estudante  
From ESTUDANTE  
Where Codigo_Estudante = 6;
```

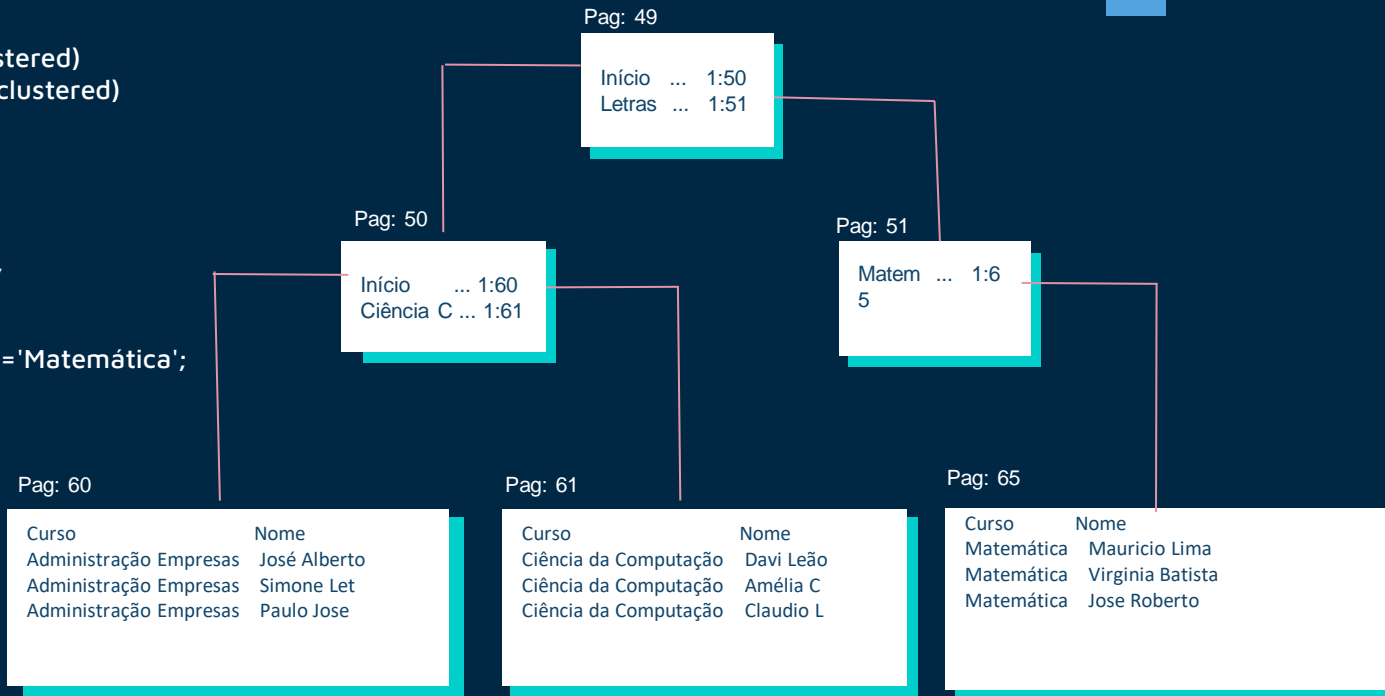


Cenário: Índice Nonclustered

TABELA com 5 colunas:

- Codigo_Estudante (Clustered)
- Curso_Estudante (Nonclustered)
- Nome_Estudante
- Idade_Estudante
- Genero_Estudante

```
Select Curso_Estudante,  
       Nome_Estudante  
From ESTUDANTE  
Where Curso_Estudante = 'Matemática';
```



LINKED SERVER

04

LINKED SERVER

Um **Linked Server** é uma funcionalidade no SQL Server que permite a criação de uma conexão para acessar e consultar dados de fontes de dados remotas, incluindo outros servidores SQL, bancos de dados de terceiros e fontes de dados externas. Isso possibilita a integração de dados de diversas origens, permitindo que você execute consultas distribuídas e operações de recuperação de dados de diferentes fontes, tudo a partir de uma única instância do SQL Server.

LINKED SERVER

Vantagens do Linked Server:

1. **Acesso a Dados Distribuídos:** A principal vantagem do Linked Server é a capacidade de acessar dados de fontes distribuídas, sejam outros servidores SQL Server ou fontes de dados externas. Isso permite a integração de dados de várias origens em um único ambiente.
2. **Facilidade de Consulta Distribuída:** O Linked Server simplifica a execução de consultas distribuídas, permitindo que você combine dados de diferentes fontes em consultas SQL, facilitando a análise e relatórios abrangentes.
3. **Centralização de Dados:** O Linked Server permite centralizar o acesso a dados de várias fontes em um único servidor SQL, simplificando o gerenciamento e a consulta de informações.
4. **Recursos de Negócios:** É útil para empresas que precisam acessar dados de sistemas legados ou fontes externas para suportar processos de negócios.

LINKED SERVER

Desvantagens do Linked Server:

1. **Problemas de Segurança:** A configuração incorreta do Linked Server pode levar a problemas de segurança, como acesso não autorizado, exposição de dados confidenciais e vulnerabilidades de injeção de SQL.
2. **Complexidade de Configuração:** Configurar um Linked Server pode ser um processo complexo, exigindo conhecimento avançado e configurações precisas. Erros na configuração podem levar a problemas de desempenho e segurança.
3. **Desempenho Inconsistente:** O desempenho de consultas distribuídas por meio de um Linked Server pode ser afetado pela latência da rede e pela eficiência das consultas, o que pode resultar em desempenho inconsistente.
4. **Manutenção Complexa:** A manutenção de Linked Servers, especialmente em ambientes com muitos servidores e fontes de dados, pode ser desafiadora e requer atualizações periódicas para manter a integridade e a segurança.
5. **Dependência de Conectividade de Rede:** O funcionamento do Linked Server depende da conectividade de rede, o que pode causar problemas se a rede apresentar interrupções.

LINKED SERVER

Para habilitar um Linked Server no SQL Server:

1. Abra o SQL Server Management Studio (SSMS).
2. Acesse a pasta "Server Objects" no Object Explorer.
3. Clique com o botão direito em "Linked Servers" e escolha "New Linked Server".
4. Configure as propriedades, incluindo nome, tipo, opções de segurança e servidor remoto.
5. Teste a conexão e, se bem-sucedido, crie o Linked Server.
6. Use o Linked Server para executar consultas distribuídas e acessar dados no servidor remoto.
7. Certifique-se de considerar a segurança e o desempenho ao fazê-lo.

Para habilitar a conexão com arquivos do Excel:

1. Baixe o Microsoft Access Database Engine Redistributable do site da Microsoft, escolhendo a versão compatível com a arquitetura do SQL Server (32 bits ou 64 bits).
2. Execute o instalador e siga as instruções.
3. Reinicie o serviço do SQL Server.

LINKED SERVER

Exemplo #1:

Consultando Google Maps

Neste exemplo, é realizada uma consulta direta à API XML do Google Maps usando o nome da cidade como entrada, a qual retorna informações adicionais e suas coordenadas geográficas.



LINKED SERVER

Exemplo #2:

Consultando bancos remotos

Neste exemplo, realizamos uma conexão remota com dois bancos de dados diferentes: um para selecionar estados da região Sul e outro para os estados da região Sudeste do Brasil. No final, combinamos e listamos os resultados ordenados por nome.



LINKED SERVER

Exemplo #3:

Consultando fontes múltiplas

Neste exemplo, listamos estados da região Nordeste de um banco local e os relacionamos com um banco remoto para filtrar estados brasileiros. Por fim, unimos esses resultados com estados estrangeiros de uma planilha do Excel e os ordenamos por país.



The background is a dark blue field decorated with various geometric elements. It includes several thin white vertical lines of varying lengths. Scattered throughout are squares of different sizes and colors: some are solid pink, some are solid teal, and others are outlined in orange or teal. These elements are positioned around the central text, creating a modern, minimalist aesthetic.

OBRIGADO