

Curso Superior de Tecnologia em Banco de Dados

Administração em Banco de Dados

Aula 6

Gerenciamento de Files e Filegroups

Particionamento de Tabelas



Agenda

- **Gerenciando de Files e Filegroups**
 - Armazenamento de Arquivos
 - Tipos de Arquivos
 - Tipos de Filegroups
- **Particionamento de Tabelas**
 - Partitioned Tables
 - Partition Functions
 - Partition Scheme

Armazenamento de Arquivos

- Dentre várias características dos bancos de dados relacionais, a garantia da persistência de dados é uma das mais importantes
- Outra tão importante quanto é a capacidade de escrita e leitura em velocidades compatíveis, garantindo o bom desempenho do sistema
- Entre os vários dispositivos de um servidor (cpu, memória, placas de rede, ...) frequentemente os discos se tornam gargalos do sistema

Armazenamento de Arquivos

- Em qualquer SGBD é fundamental o entendimento do armazenamento de arquivos em dispositivos que ofereçam proteção contra falhas e garantam alta performance de desempenho, muitas vezes, distribuindo os dados fisicamente em dispositivos distintos
- Em 1988 foi proposto a criação de um subsistema de armazenamento composto por vários discos individuais, com a finalidade de ganhar segurança e desempenho e chamado RAID ("Redundant Array of Independent Disks" - Conjunto Redundante de Discos Independentes)

Armazenamento de Arquivos

- Popularmente RAID são dois ou mais discos trabalhando simultaneamente par um mesmo fim
- Níveis de RAID são maneiras de como combinar estes discos de forma que ofereçam performance, segurança ou a combinação destas características
- Existem vários níveis de RAID (0, 1, 2, 3, 4, 5, 6) e combinação destes níveis (01, 10, 50, 100). Os mais comuns serão mostrados a seguir

Armazenamento de Arquivos – RAIDS



Standalone



Cluster



Hot swap



RAID 0



RAID 1



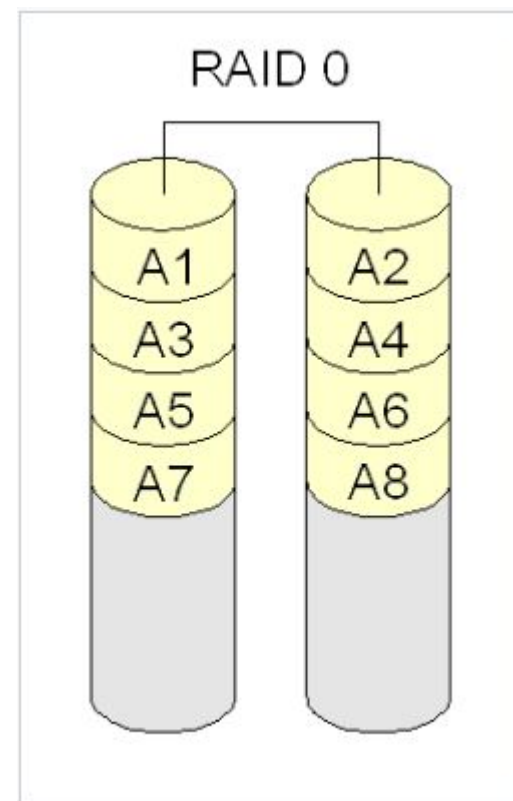
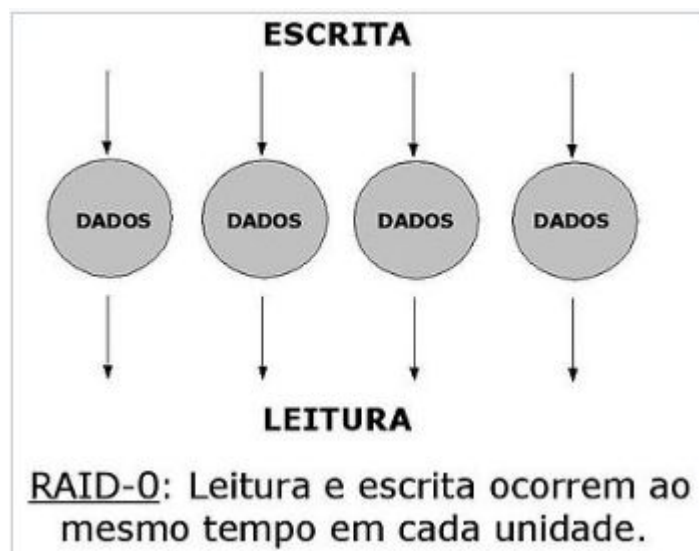
RAID 5



RAID 0+1

Armazenamento de Arquivos – RAID 0

- RAID 0 é um sistema de dois ou mais discos cujo principal objetivo é deixar o sistema mais rápido através da divisão de dados entre os discos (Striping)



Armazenamento de Arquivos – RAID 0

- Vantagem

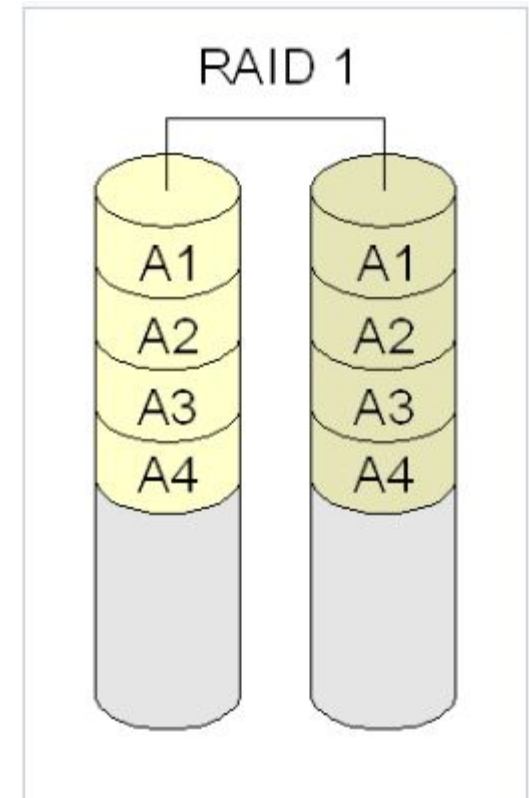
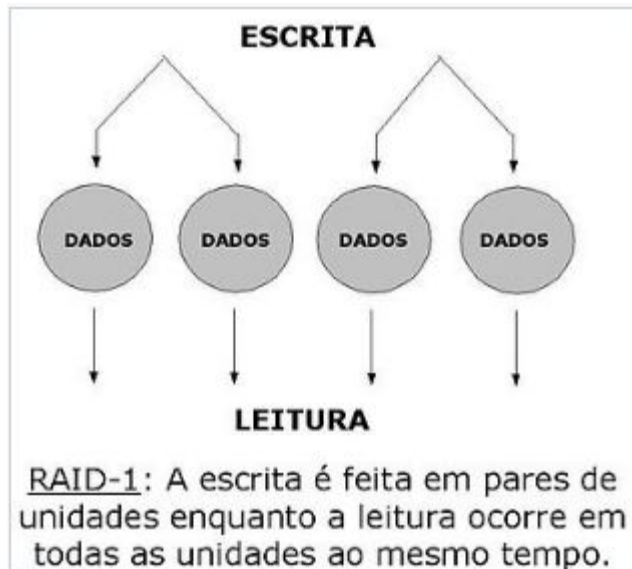
- ✓ Performance - Acesso mais rápido para leitura e escrita
- ✓ Baixo custo para expansão

- Desvantagem

- ✓ Caso algum dos setores de algum dos discos tenha perda de informações, o mesmo arquivo que está dividido entre os mesmos setores dos demais discos não conseguirá ser lido, pois uma parte do arquivo foi corrompida, ou seja, caso algum disco falhe, não temos como recuperar os dados

Armazenamento de Arquivos – RAID 1

- RAID 1 (Mirror) implementa o espelhamento entre discos. São necessários no mínimo dois discos ou mais desde que sejam utilizados sempre em pares



Armazenamento de Arquivos – RAID 1

- Vantagem

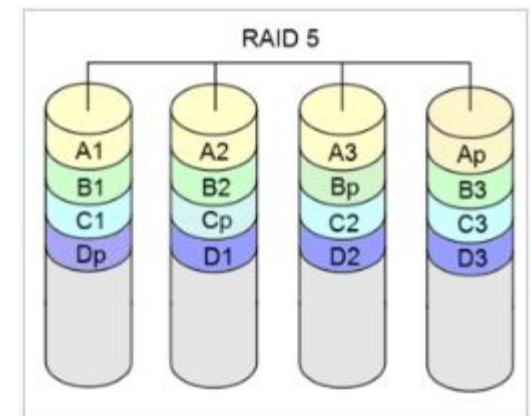
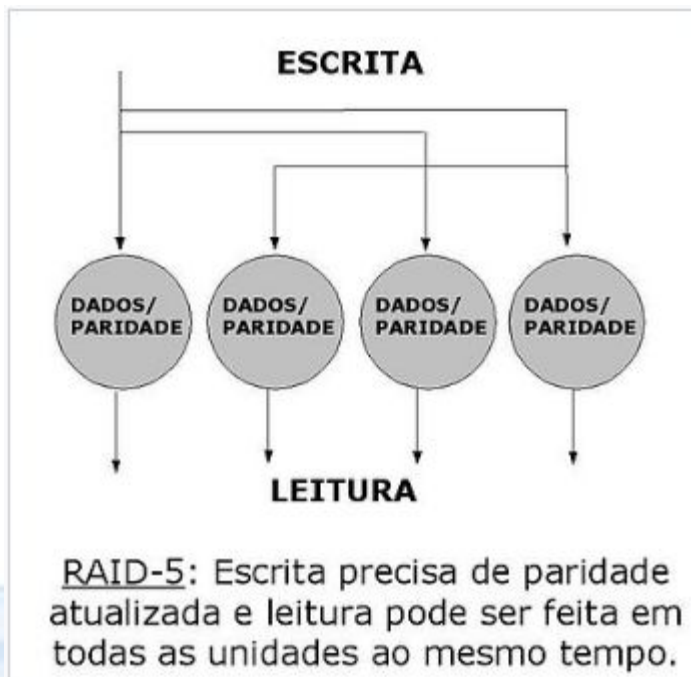
- ✓ Segurança - Caso algum setor de um disco tenha falha, os mesmos dados estão armazenados no disco espelhado, conseguindo a recuperação dos dados

- Desvantagem

- ✓ Custo relativamente alto comparado ao RAID 0. Gastamos dois discos para manter a informação referente a 1 disco.
- ✓ Aumento no tempo de escrita

Armazenamento de Arquivos – RAID 5

- RAID 5 é um dos sistemas mais utilizados, podendo ser construído a partir de 3 discos e implementa informações de paridade (um valor calculado que pode ser usado para reconstruir dados após uma falha) para os dados que estão distribuídos no array de discos



Armazenamento de Arquivos – RAID 5

- Vantagem

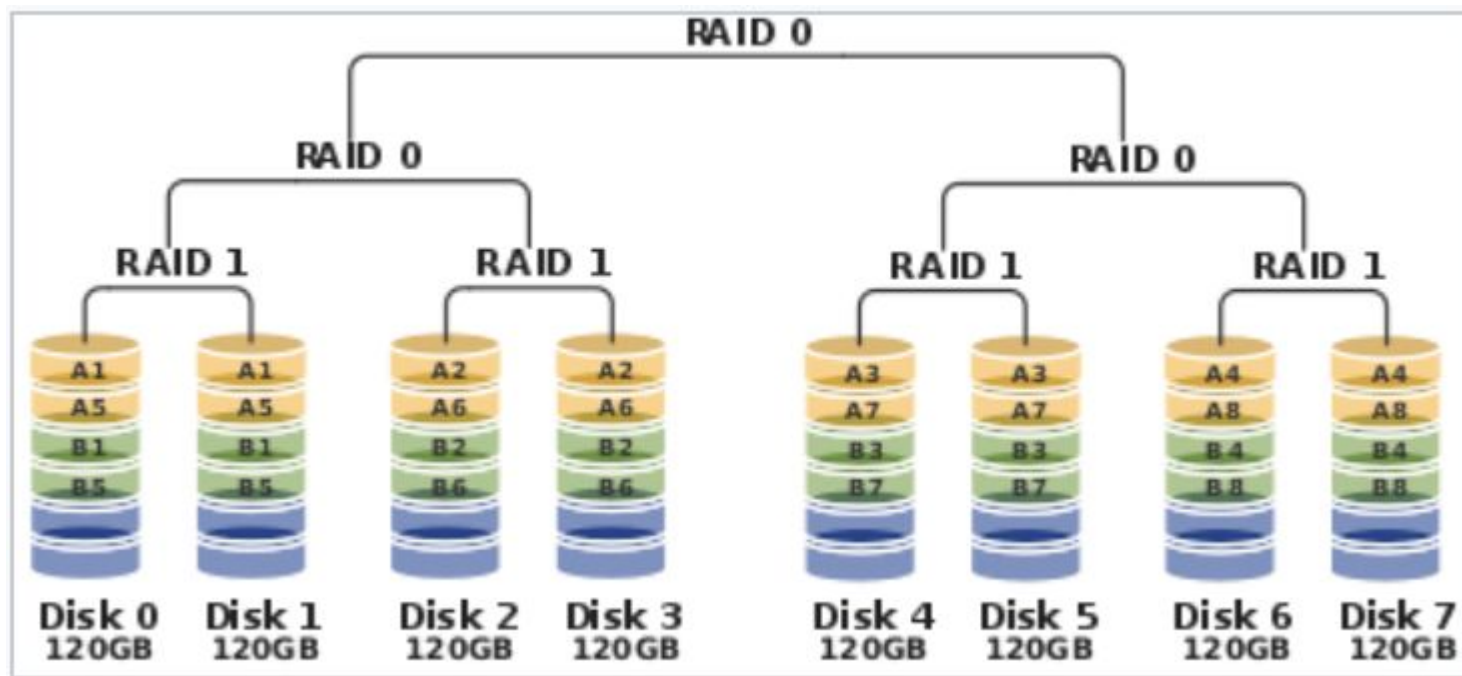
- ✓ Leitura muito rápida (porém escrita não tão rápida)
- ✓ Custo médio – Gastamos um terço do disco para informações de paridade

- Desvantagem

- ✓ Sistema complexo de controle dos HDs

Armazenamento de Arquivos – RAIDS

- Os níveis de RAID também podem ser combinados e podendo ter complexos sistemas de armazenamentos



O que são tipos de Arquivos ?

Tipo de Arquivos	Extensão	Description
Primary Files	.mdf	O arquivo primário de dados contem as informações de startup para o banco de dados e mapeamento para os outros arquivos no banco de dados. É um arquivo obrigatório no database.
Secondary Files	.ndf	Arquivos secundários são opcionais, definidos pelo usuário. Podem ser usados para espalhar os dados em vários discos, colocando cada arquivo em uma unidade de disco diferente.
Transaction logs	.ldf	Arquivos de logs transacionais mantém TODAS as informações necessárias para a recuperação de um database. É um arquivo obrigatório no database.

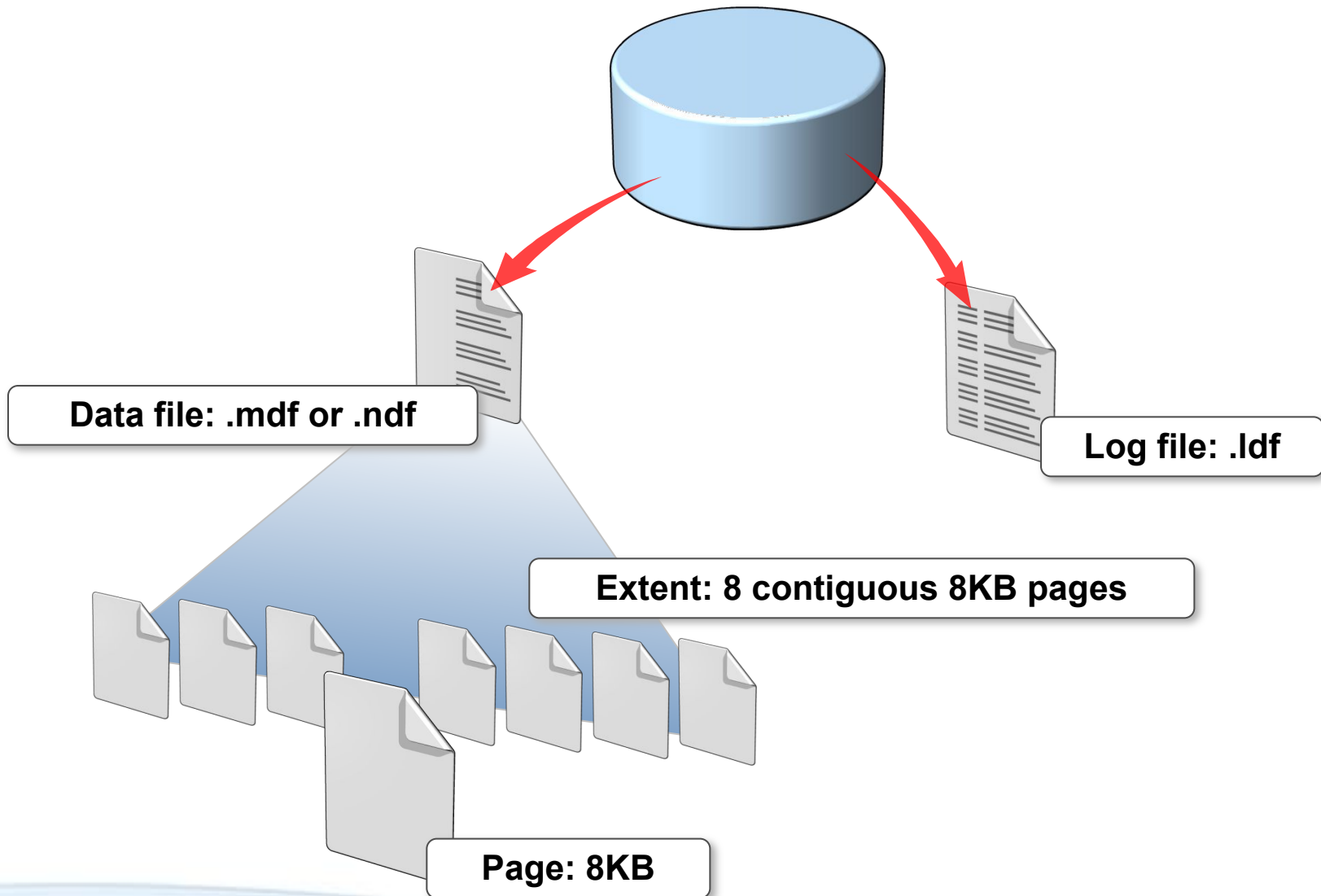
O que são Filegroups?

- Filegroups são coleções de arquivos onde os dados do banco de dados são armazenados
- Todo banco de dados possui no mínimo um filegroup que pode ser formado por um ou mais arquivos
- O filegroup é um objeto lógico que mapeia arquivos físicos de dados
- São essenciais para melhorar o desempenho do banco de dados, permitindo a distribuição dos dados, em arquivos espalhados por vários discos

Filegroups

- Todo database possui obrigatoriamente o PRIMARY FILEGROUP, que obrigatoriamente mapeia seu PRIMARY DATA FILE (MDF). Esta obrigatoriedade é devido a construção dos metadados, pois não importa o número de filegroups ou arquivos, serão deixados no MDF.
- Demais arquivos criados nos databases, denominamos SECONDARY DATA FILE (NDF) e podemos direcionar fisicamente objetos como tabelas, índices, ...
- O log transacional (LDF) não participa de FILEGROUPS, apenas os arquivos de dados (MDF e NDF)

Files e Filegroups



Considerações sobre Arquivos

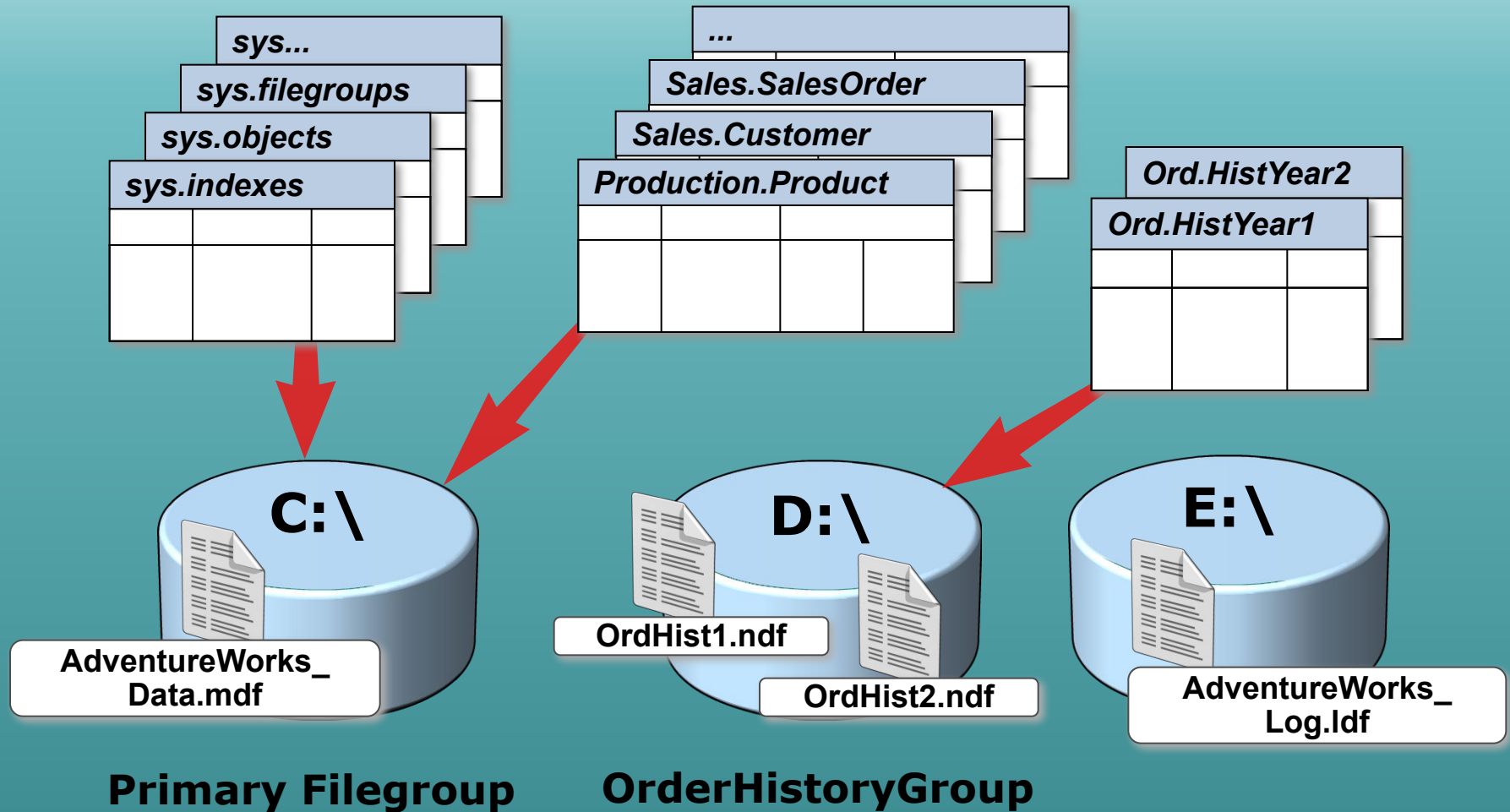
File Type	Consideration
Primary and secondary files	<ul style="list-style-type: none">• Se existirem tabelas que são usadas com frequência em conjunto, devem ser colocadas em filegroups e unidades físicas separadas
tempdb	<ul style="list-style-type: none">• Coloque o database tempdb em um subsistema de unidade de disco rápida para garantir um bom desempenho
Transaction Logs	<ul style="list-style-type: none">• Crie o log transaccional em discos físicos separados ou em um RAID array

Planejamento de Discos e Armazenamento de Arquivos

As considerações mais importantes são o número de discos disponíveis e a velocidade das unidades envolvidas

Files e Filegroups

AdventureWorks Database



Considerações para Filegroups

Filegroups podem ser criados juntamente com o banco de dados ou posteriormente, quando adicionarmos arquivos

- ✓ Files não podem ser movidos para diferentes filegroups
- ✓ Um arquivo não pode participar em mais de um filegroup
- ✓ Filegroups podem conter somente DATA FILES
- ✓ Crie filegroups em discos físicos diferentes
- ✓ Coloque tabelas que participam de queries em diferentes filegroups
- ✓ Coloque tabelas com uso massivo em diferentes filegroups

Demonstração

```
-- Gerenciando FILEGROUP e FILE
-- Drop Database DBAula
CREATE DATABASE DBAula
ON PRIMARY (NAME = 'DBAula_Data', FILENAME = 'C:\BD\MDF\DBAula_Data.mdf',
SIZE = 25 MB, MAXSIZE = 40 MB, FILEGROWTH = 10 MB)
LOG ON (NAME = 'DBAula_log', FILENAME = 'C:\BD\LDF\DBAula_Log.ldf',
SIZE = 5 MB, FILEGROWTH = 10 %)

-- Adicionando FILEGROUPs
alter database DBAula add Filegroup NovaAula

-- Adicionando File
alter database DBAula
add file (name = 'NovaAula_Data', filename = 'C:\BD\MDF\NovaAula_Data.ndf', size = 5MB, filegrowth=10%)
to Filegroup NovaAula

-- Removendo Arquivos File (vazio)
alter database DBAula remove file NovaAula_Data

-- Adicionando Files
alter database DBAula
add file
(name = 'NovaAula_Data1', filename = 'C:\BD\MDF\NovaAula_Data1.ndf', size = 5MB, filegrowth=10%)
, (name = 'NovaAula_Data2', filename = 'C:\BD\MDF\NovaAula_Data2.ndf', size = 5MB, filegrowth=10%)
to Filegroup NovaAula

-- Removendo Arquivos File (vazio) e Filegroup
alter database DBAula remove file NovaAula_Data1
alter database DBAula remove file NovaAula_Data2
alter database DBAula remove Filegroup NovaAula
```

Demonstração

-- Direcionando objetos

CREATE DATABASE DBAula

ON PRIMARY (NAME = 'DBAula_Data1', FILENAME = 'C:\BD\MDF\DBAula_Data1.mdf', SIZE = 5 MB, FILEGROWTH = 10 MB),

FILEGROUP CORPORATIVO DEFAULT (NAME = 'DBAula_Data2', FILENAME = 'C:\BD\MDF\DBAula_Data2.ndf', SIZE = 20 MB, FILEGROWTH = 10 %),

FILEGROUP HISTORICO

(NAME = 'DBAula_Data3', FILENAME = 'C:\BD\MDF\DBAula_Data3.ndf', SIZE = 10 MB, FILEGROWTH = 5 MB),

(NAME = 'DBAula_Data4', FILENAME = 'C:\BD\MDF\DBAula_Data4.ndf', SIZE = 10 MB, FILEGROWTH = 5 MB)

LOG ON

(NAME = 'DBAula_log', FILENAME = 'C:\BD\LDF\DBAula_Data4.mdf', SIZE = 10 MB, FILEGROWTH = 10 %)

use DBAula

-- Criando Objetos no filegroup DEFAULT

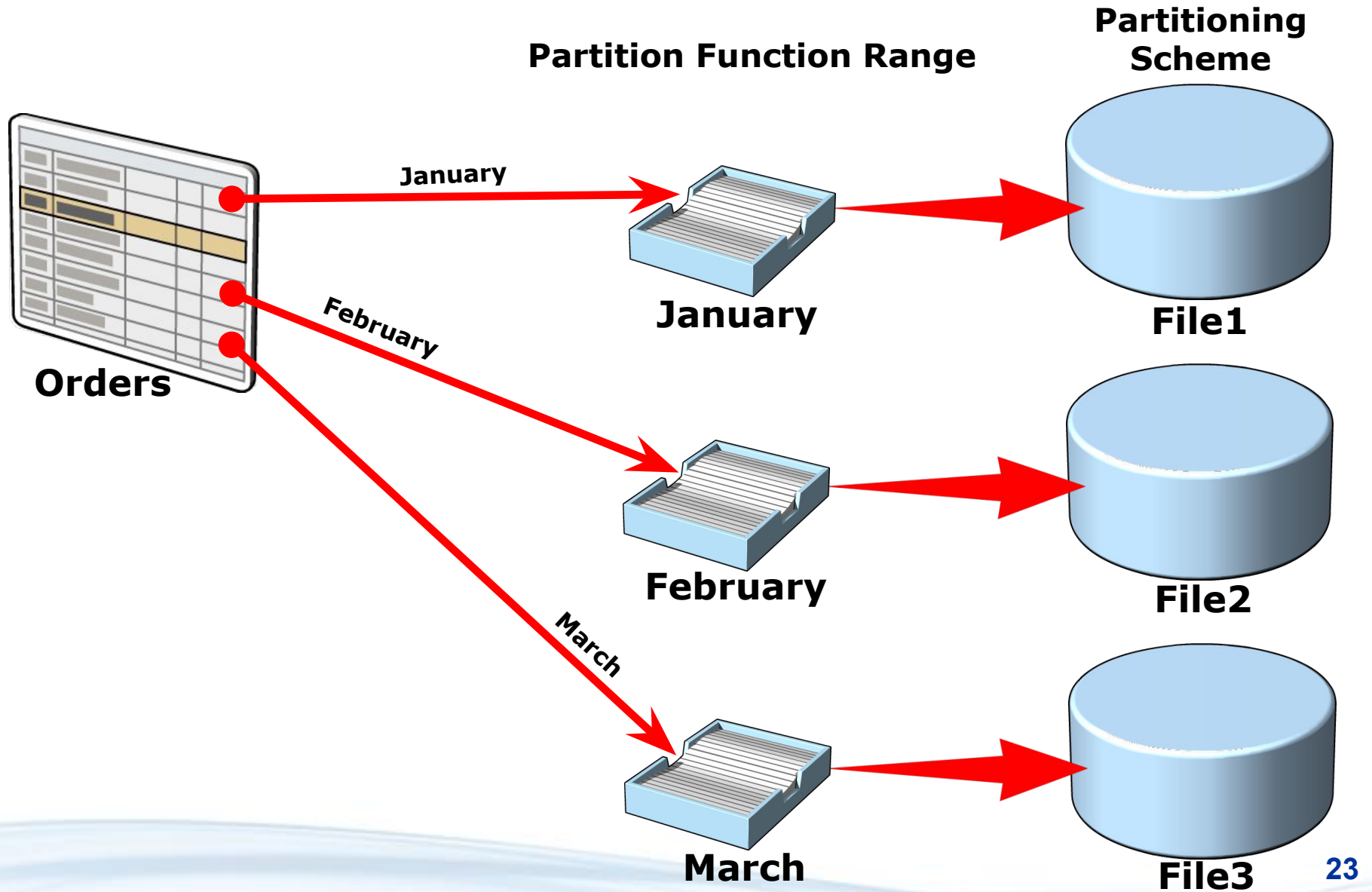
create table Cliente (idCliente int identity primary key, nome varchar(20), sobrenome varchar(20))

-- Criando Objetos no filegroup desejado

create table Produto (idProduto smallint identity primary key, nome varchar(50), descricao varchar(200))
on corporativo

create table Venda (idVenda int identity primary key, data datetime default getdate(), idCliente int
foreign key references Cliente(idCliente), idProduto smallint foreign key references Produto(idProduto))
on Historico

O que são Partições?



O que são Tabelas Particionadas ?


- Divisão de tabelas em unidades
- Torna tabelas e índices grandes mais gerenciáveis
- Acessar subconjuntos de dados de forma mais rápida e eficiente, mantendo a integridade dos dados
- As unidades divididas podem ser distribuídas em mais de um filegroup num banco de dados
- Os dados são particionados horizontalmente, de modo que grupos de linhas são mapeados em partições individuais
- A tabela ou índices são tratados como uma única entidade lógica em consultas ou atualizações dos dados

O que são Tabelas Particionadas ?

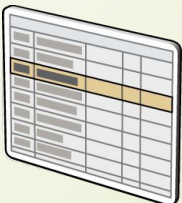
- Todas as propriedades e recursos associados à criação da tabelas são mantidos (constraints, defaults, identity, triggers, ...)
- Aplicação e eficiência para tabelas grandes onde os dados são solicitados de diferentes maneiras
- Aplicação quando consultas ou atualizações no objeto original não estão funcionando como planejado (lentidão) ou custos de manutenção excedem os períodos de manutenção pré-definidos.

Partitioned Tables


Dados são particionados horizontalmente, por faixas (range)




< 2003



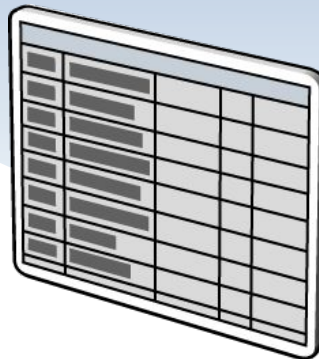
2003



2004



>= 2005



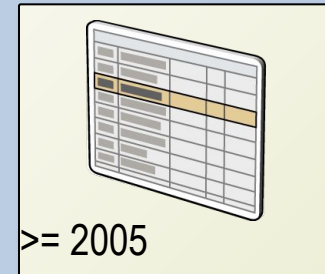
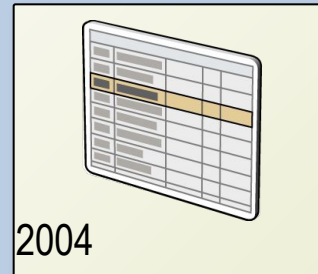
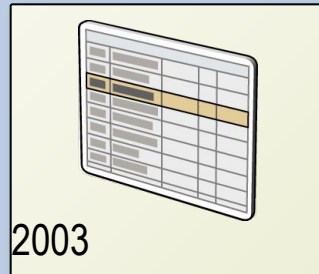
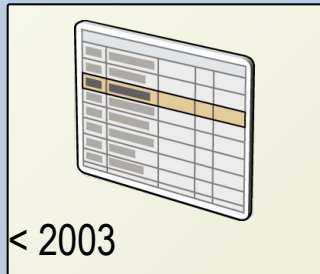
Sales.Orders

O que são Funções de Partição

- Especifica como uma tabela ou índice será particionado
- Mapeia o domínio em um conjunto de partições
- Na criação da função, especificamos o número de partições, a coluna de particionamento e o intervalo de valores da coluna para cada partição
- Trabalham com valor limite, onde configuramos em que lado da partição, quando ocorrer o determinado valor, será armazenado (ESQUERDA ou DIREITA de cada partição)

Partition Function

- Funções de Partição definem limites (boundaries) das partições
- Valores limites são atribuídos a **ESQUERDA** ou **DIREITA**



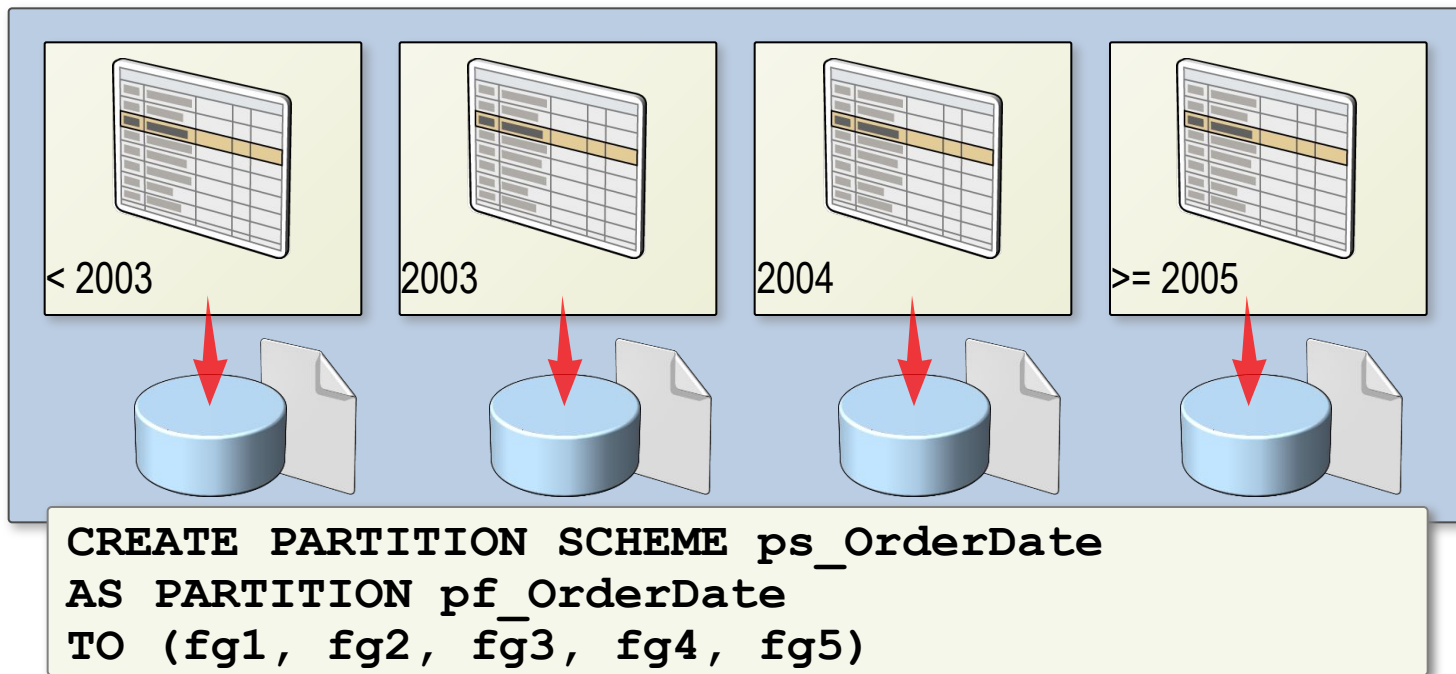
```
CREATE PARTITION FUNCTION pf_OrderDate (datetime)
AS RANGE RIGHT
FOR VALUES ('01/01/2003', '01/01/2004', '01/01/2005')
```

O que são Partition Scheme ?

- Mapeia as partições produzidas por uma função de partição para um conjunto de filegroups definidos
- Um esquema de partição pode usar apenas uma função de partição, no entanto, uma função de partição pode participar em mais de um esquema de partição
- Na criação da função, especificamos o número de partições, a coluna de particionamento e o intervalo de valores da coluna para cada partição
- No planejamento do particionamento de tabelas, definimos o número de filegroups para colocar as partições, a função de partição que definirá como as linhas da tabela serão mapeadas para o conjunto de partições e o esquema de partição que mapeia cada partição definida para o respectivo filegroup

Partition Scheme

- Um Partition Schema atribui partições a filegroups
- Um “próximo” filegroup pode ser definido e configurado para novas linhas



Demonstração

```
USE master
-- Drop Database DBAula
CREATE DATABASE dbaula ON PRIMARY
(NAME = 'DBAula_Data', filename = 'C:\BD\MDF\DBAula_Data.mdf', size = 25 mb, maxsize = 40 mb, filegrowth =
10 mb)
log ON (NAME = 'DBAula_log', filename = 'C:\BD\LDF\DBAula_Log.ldf', size = 5 mb, filegrowth = 10 %)
go

-- Adicionando FILEGROUPs
ALTER DATABASE dbaula ADD filegroup [FG_Historico]
go

USE dbaula
go

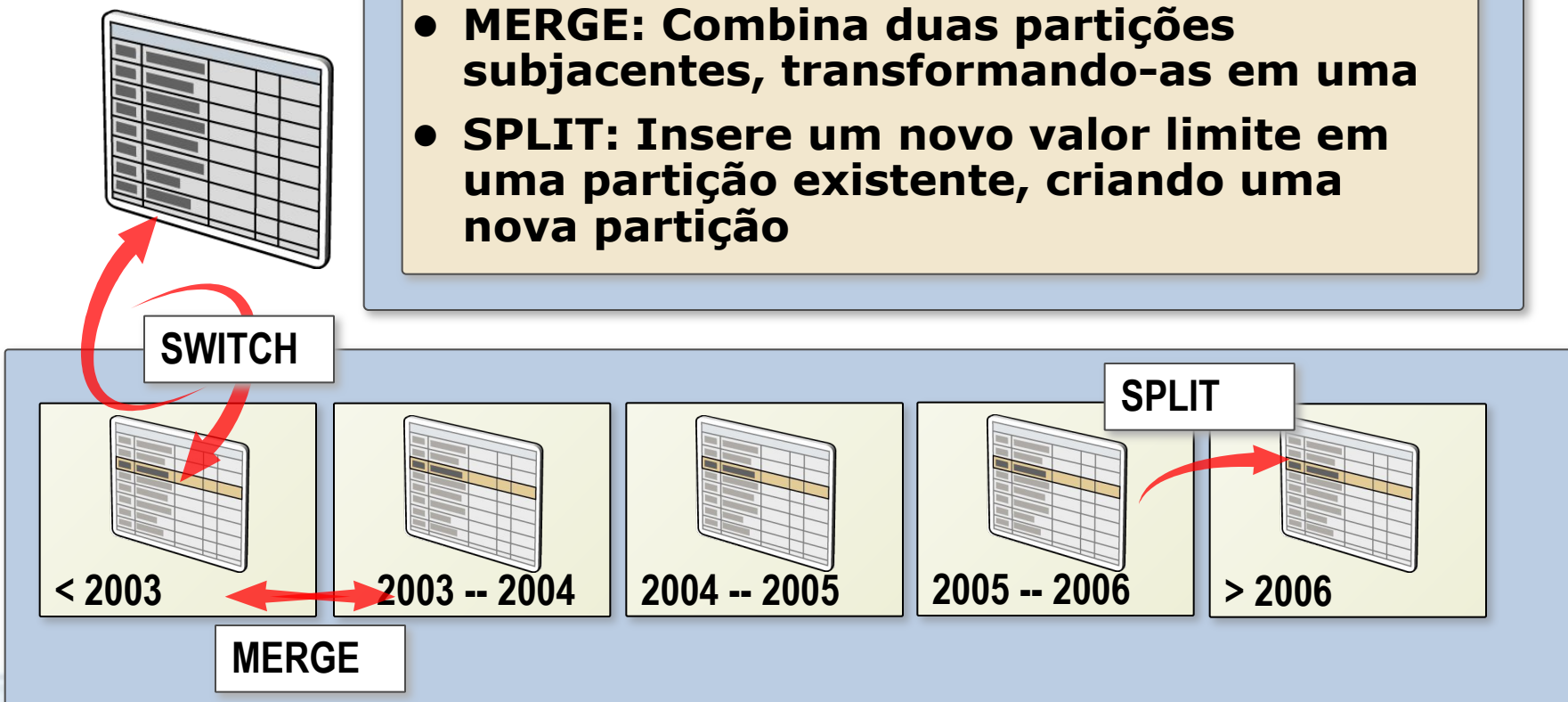
--Crio função de partição: responsável por saber onde dividir o dado.
CREATE partition FUNCTION [PF_HISTORICO](datetime) AS range LEFT
FOR VALUES ( '2019-01-01 00:00:00.000')

--Crio schema de partição: responsável por saber onde alocar os dados
CREATE partition scheme [PS_HISTORICO]
AS partition pf_historico TO ( [FG_HISTORICO], [PRIMARY])

--Crio tabela particionada pelo schema + função
CREATE TABLE tableaula
(
    id          INT NOT NULL IDENTITY(1, 1),
    datahora    DATETIME NOT NULL,
    CONSTRAINT pk_consulta PRIMARY KEY ( id, datahora ) ON ps_historico ( datahora )
)
go
```

O que operações podem ser feitas com Dados Particionados?

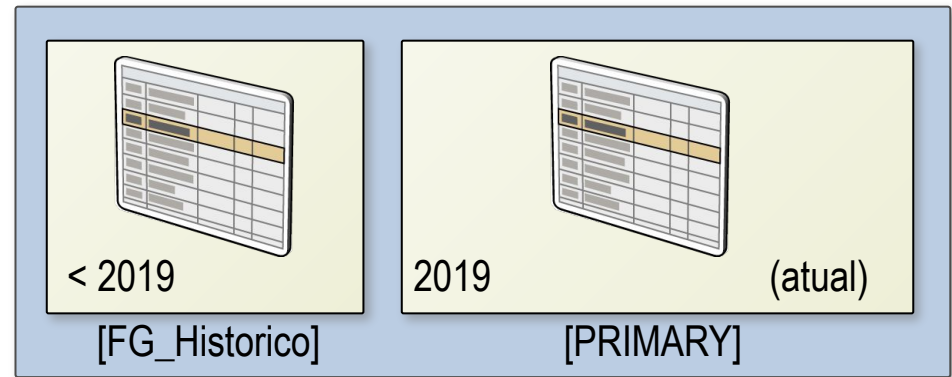
- **SWITCH:** Manipula dados de uma tabela populada ou partição, para uma tabelas ou partição vazia
- **MERGE:** Combina duas partições subjacentes, transformando-as em uma
- **SPLIT:** Insere um novo valor limite em uma partição existente, criando uma nova partição



Demonstração

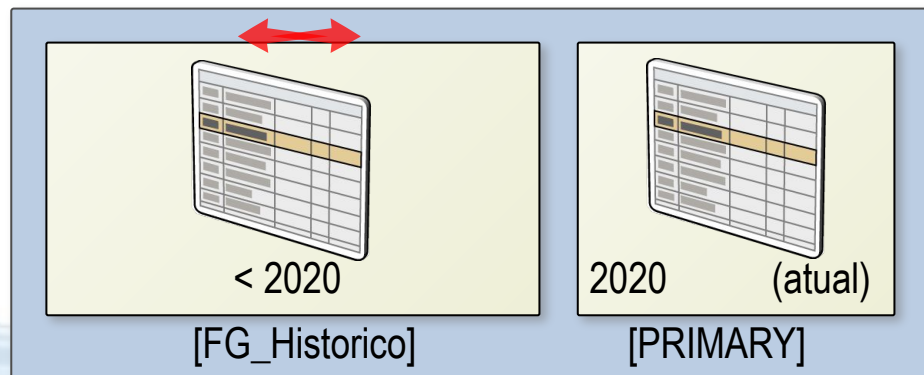
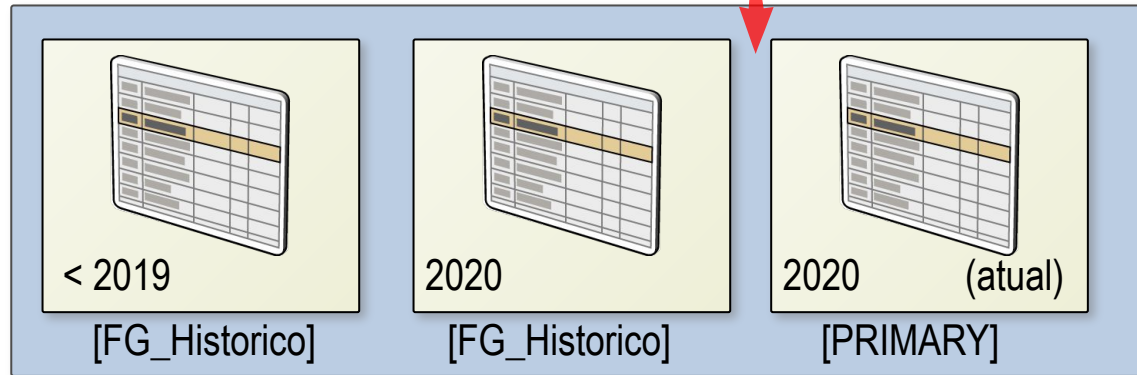
--SPLIT

```
ALTER partition scheme ps_historico next used fg_historico;  
ALTER partition function pf_historico() split range ('2020-01-01 00:00:00.000');
```



-- MERGE

```
ALTER partition function pf_historico() merge range ('2019-01-01 00:00:00.000');
```



Demonstração

--Validação da separação dos dados.

```
SELECT p.partition_number,  
       fg.NAME,  
       p.rows  
FROM   sys.partitions p  
       INNER JOIN sys.allocation_units au  
                ON au.container_id = p.hobt_id  
       INNER JOIN sys.filegroups fg  
                ON fg.data_space_id = au.data_space_id  
WHERE  p.object_id = Object_id('TableAula')
```

Curso Superior de Tecnologia em Banco de Dados

Obrigado!

Prof. Gustavo Maia
gustavo.maia@faculdadeimpacta.com.br

Material criado e cedido por
Prof. Sand Onofre

