Banco de Dados

Módulo Intermediário

Microsoft®
SQL Server®

VISÃO GERAL

SQL Server

- ***Segurança***
  - ***Criptografia nativa***
  - ***Exemplo:***
    - *Se você desejar **tornar seguros** os **dados sensíveis** dos **funcionários**, utilizando **criptografia** em **nível** de **coluna**, poderia **criptografar uma única coluna** em uma **tabela***

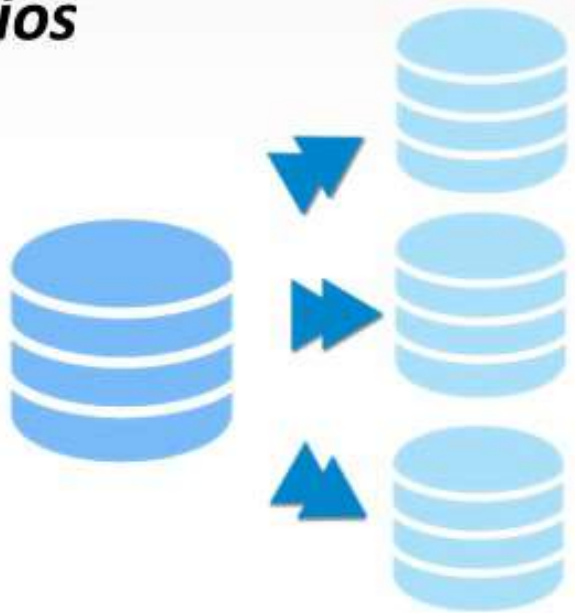
  - ***TDE** (Transperent Data Encryption)*
    - ***Permite criptografar** um **banco** de **dados inteiro sem afetar** o modo como os **clientes** e **aplicativos acessam** os **dados***
    - *Se alguém **quisesse violar** sua **segurança** de **rede** e **obter** uma **cópia** de um **arquivo** de **dados** ou de um **arquivo** de **backup**, a única maneira de **acessar** os **dados** seria com uma **chave** de **criptografia***

Replicação

- ***Snapshot***
  - ***Forma** mais **simples** de **replicação***
  - ***Extrai** um **snapshot** dos **dados** **periodicamente** distribuindo aos **servidores***
  - *Usada para **mover dados** em **intervalos mais longos**, como **diariamente** (ou a cada noite)*
  - *Embora esse **método** seja **eficiente**, muitas vezes é **insuficiente** para **satisfazer** as **altas demandas** dos **usuários** por **dados** em **tempo real***
  - *Se for necessário um **desempenho** (**troughput**) **mais alto**, muitas vezes os **usuários** utilizam **replicação transactional***

Replicação

- ***Replicação Transacional***
  - ***Envia*** *as* ***alterações feitas*** *nos* ***dados*** ***continuamente****,* *à* ***medida*** *que elas* ***acontecem***
  - *É utilizada em uma* ***topologia*** *servidor-para-servidor, onde um* ***servidor*** *é a* ***fonte*** *de* ***dados*** *e o outro é* ***usado*** *como* ***cópia*** *de* ***backup*** *ou para* ***relatórios***



Replicação

- ***Snapshot*** *e* ***Transactional***
  - *Os* ***dois tipos*** *de* ***replicação*** *são* ***movimentações*** *de* ***dados*** ***unidirecionais***
  - *E se* ***você precisar*** *de* ***movimentação*** ***bidirecional?***
    - ***Exemplo:***
      - ***Imagine*** *que* ***você tenha usuários móveis*** *que* ***trabalham*** ***offline***
      - ***Enquanto*** *estão* ***offline****,* *eles* ***inserem dados*** *em um* ***BD residente*** *em uma* ***instância*** *em* ***execução*** *nos seus* ***laptops***
      - *O que* ***acontece*** *quando eles* ***retornarem*** *para o* ***escritório*** *e se* ***conectarem*** *na* ***rede?***
      - *Nesse cenário, a* ***instância local*** *será* ***sincronizada*** *com o* ***BD principal***

Replicação

- ***Snapshot** e **Transactional***
  - *E se **você precisar** de **movimentação bidirecional**?*
    - ***Exemplo:***
      - *A **replicação** usando **merge** (**mesclagem**) **moverá** as **transações** entre o **editor** (**publisher**) e o **assinante** (**subscriber**) desde a **última vez** que a **sincronização ocorreu***

Replicação

- ***SQL Server Agent***
  - ***Notifica*** *os* ***responsáveis*** *no* ***caso*** *de* ***uma falha*** *na* ***job***
  - ***Permite*** *a* ***configuração*** *de* ***operadores*** *e* ***alertas*** *(uma pessoa e um endereço de e-mail)*
  - ***Uma vez configurado*** *um* ***operador****, podemos* ***enviar notificações*** *ou* ***alertas*** *para uma* ***determinada pessoa****, quando um* ***job*** *tiver* ***sucesso****,* ***terminar*** *ou* ***falhar***

SQL Server

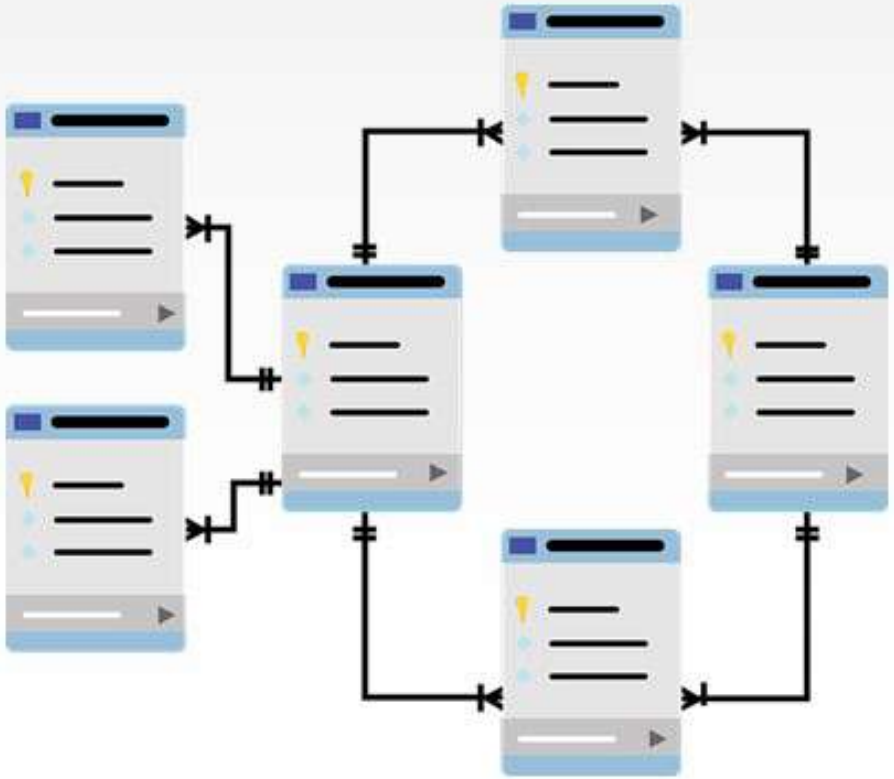
- ***Edição: Enterprise***
  - ***Considerada** a **edição premium** do **SQL Server***
  - ***Completa** (contém **todos** os **recursos disponíveis**)*
  - ***Apresenta** uma **solução** de **centro** de **dados completa**, que **suporta** um **alto nível** de **workloads** de **missão crítica**, **excelente desempenho**, **virtualização** e **recursos** de **business intelligence (BI)***

SQL Server

- ***Edição: Business Intelligence***
  - ***Concentra-se*** *na* ***implementação*** *de* ***soluções amplas, voltadas*** *para* ***BI***
  - ***Permite*** *que as* ***organizações construam, implantem*** *e* ***gerenciem soluções altamente escalonáveis*** *de forma* ***eficiente*** *e* ***eficaz***
  - *Ao* ***acessar dados****, os* ***usuários finais*** *terão uma* ***experiência baseada*** *em* ***navegador*** *que os* ***permitirá decompô-los*** *e* ***analisa-los***

SQL Server

- ***Edição: Standard***
  - ***Considerada tão robusta*** *quanto a* ***edição Enterprise*** *ou* ***BI***
  - *Apresenta* ***vários recursos relevantes****, além de* ***recursos básicos*** *de* ***gerenciamento*** *de* ***dados*** *e* ***BI*** *(menor escala)*
  - ***Caso*** *sua* ***necessidade*** *seja a* ***nível departamental*** *ou* ***uma pequena organização****, esta é a* ***versão apropriada***



PROJETO DE BD

BD

- ***Objetivos***
  - ***Entender*** *os* ***requisitos*** *e* ***funções*** *de cada* ***BD*** *de* ***sistema***
  - ***Entender*** *a* ***estrutura*** *de um* ***BD SQL Server***
  - ***Criar*** *um* ***BD***
  - ***Adicionar*** *e* ***alterar filegroups***
  - ***Adicionar arquivos filegroups***
  - ***Desanexar*** *e* ***anexas BDs***
  - ***Entender*** *os* ***modelos*** *de* ***recuperação*** *de* ***BDs***

BD

- ***BDs** de **Sistema***
  - *Os **BDs** de **sistema** são **criados** por **padrão** quando uma **instância** de **SQL Server** é **instalada***
  - ***Cada** um dos **bancos** de **dados** tem um **propósito específico** e é necessário para a **execução** do **SQL Server***
    - ***master***
    - ***tempdb***
    - ***model***
    - ***msdb***
    - ***resource***
    - ***distribution***

BD

- ***BDs** de **Sistema***
  - ***master:***
    - ***Principal BD** do **sistema***
    - *Sem ele, o **SQL Server não** pode **iniciar***
    - ***Contém** as **informações** mais **importantes** sobre os **objetos dentro** da **instância** como: **bancos** de **dados**, **alwaysON**, **espelhamento** de **banco** de **dados**, **configurações**, **logins**, **resource governor** (administrador de recursos) e **endpoints***

- ***BDs** de **Sistema***
  - ***Exemplo (01):***
    - ***Listar todos** os **bancos** de **dados** em uma **instância** do **SQL Server***

```
USE master;

SELECT *
FROM sys.master_files;
```

name	physical_name
master	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\master.mdf
mastlog	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\mastlog.ldf
tempdev	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\tempdb.mdf
templog	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\templog.ldf
modeldev	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\model.mdf
modellog	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\modellog.ldf
MSDBData	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\MSDBData.mdf
MSDBLog	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\MSDBLog.ldf
Aula_03_08_22	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Aula_03_08_22.mdf
Aula_03_08_22_log	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Aula_03_08_22_log.ldf
bkBank_03_08_22	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\bkBank_03_08_22.mdf
bkBank_03_08_22_log	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\bkBank_03_08_22_log.ldf
AdventureWorks2012_Data	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\AdventureWorks_Data.mdf
AdventureWorks2012_Log	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\AdventureWorks_Data_log.ldf
Academico	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Academico.mdf
Academico_log	D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Academico_log.ldf

- ***BDs** de **Sistema***
  - ***tempdb:***
    - ***Área** de **armazenamento global** para os **objetos temporários criados** pelos **processos internos** que **executam** o **SQL Server** e **objetos temporários criados** por **usuários** ou **aplicativos***
    - *Esses **objetos temporários** incluem **tabelas temporárias** e **stored procedures**, **variáveis** de **tabela**, **tabelas temporárias globais** e **cursores***
    - ***Armazena row versions** (versões do registro) de **transações read-committed** ou de **isolamento** "**snapshot**", **operações** de **índice online** e **triggers** (gatilhos) **AFTER***
    - ***Recriado sempre** que o **SQL Server** é **reiniciado***
    - ***Não utilizar** para **armazenamento** de **informações permanentes***

- ***BDs de Sistema***
  - ***model:***
    - *Um **modelo** para **todos** os **BDs criados** em **uma instância***
    - ***Usado** como um **template** sempre que **um BD** é **criado***

Se o BD model não existir ou estiver offline, o tempdb não poderá ser criado.

Isso ocorre porque, ele é criado sempre que o SQL Server é reiniciado.

Como cada BD utiliza o model como template, e o tempdb não é execução, ele deve existir para recriar o tempdb na inicialização.

BD

- ***BDs** de **Sistema***
  - ***resource:***
    - *É um **BD** **oculto**, **somente leitura** (**não** é **discutido** com **muita frequência**)*
    - ***Principal objetivo** é **melhorar** o **processo** de **atualização** de uma **versão** de **SQL Server** para a **versão seguinte***
    - ***Todos** os **objetos** de **sistema** de **uma instância** são **armazenados dentro** do **banco** de **dados** **resource***
    - ***Não** é possível **fazer** **backup** ou **restaurar***

- ***BDs** de **Sistema***
  - ***distribution:***
    - ***Somente existirá** quando for **configurado** essa **instância** como **distribuidora** para **replicação***
    - ***Antes** de **configurar** a **replicação**, é necessário aplicar essa configuração*
    - ***Todos** os **metadados** e o **histórico** dos **vários tipos** de **replicação** são **armazenados dentro desse banco** de **dados***

SQL Server

- ***Estrutura***
  - *Os **bancos** de **dados** podem ser **criados** com **muitas tecnologias** e **técnicas** distintas*
  - ***Por padrão**, todo **banco** de **dados SQL Server** é **composto** de **dois arquivos***
    - *O **arquivo** de **dados** contém **dados** e **objetos** do **banco** de **dados**, como **tabelas**, **views** (modos de exibição) e **store procedures** (procedimentos armazenados)*
    - *O **arquivo** de **log** contém **informações** que **ajudam** na **capacidade** de **recuperação** de **transações** do **banco** de **dados***

SQL Server

- ***Estrutura***

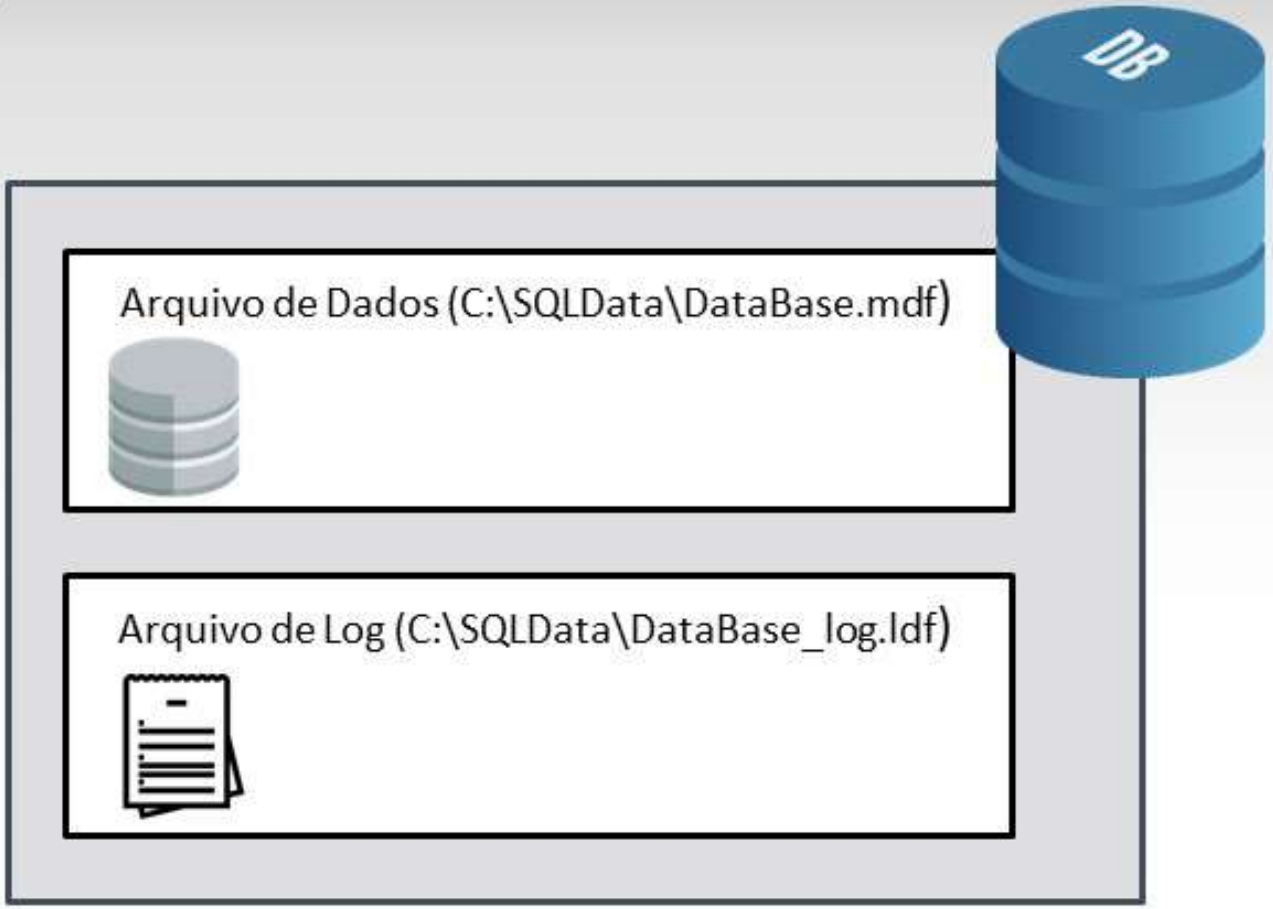


Figura 1 – A estrutura de um BD SQL Server consiste em pelo menos um arquivo de dados e um arquivo de logo

- ***Estrutura***
  - ***Exemplo (02):***
    - ***Criando** um **banco** de **dados** com **T-SQL***

```
USE master;

CREATE DATABASE bkBankAula10
ON PRIMARY
(NAME='bkBankAula10',
 FILENAME='D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\bkBankAula10.mdf',
 SIZE=10MB, MAXSIZE=20, FILEGROWTH=10%)
LOG ON
(NAME='bkBankAula10_log',
 FILENAME='D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\bkBankAula10_log.ldf',
 SIZE=10MB, MAXSIZE=200, FILEGROWTH=20%);
```

SQL Server

- ***Argumentos***
  - *nome_do_banco_de_dados é o **nome** do **banco** de **dados**, o qual deve ser **único** em **relação** à **qualquer** um dos BDs que **existam** no **momento** da **criação***
  - *ON especifica o **filegroup** e **inicia** a **seção** em que o **arquivo** de **dados** é **definido***
  - *LOG ON **inicia** a **seção** em que o **log** é **definido***
  - *Name é o **nome** do **arquivo lógico** utilizado pelo **SQL Server** ao **referenciar** o **arquivo** (assim como o nome do banco de dados, ele deve ser exclusivo)*

SQL Server

- ***Argumentos***
  - *FileName é o **caminho** no **sistema operacional** e **nome** do **arquivo**, **incluindo** a **extensão***
  - *Size especifica o **tamanho inicial** do **arquivo** em **megabytes** (**MB**), **por padrão**. Também podem ser especificados **kilobytes** (**KB**), **gigabytes** (**GB**) e **terabytes** (**TB**)*
  - *MaxSize especifica o **tamanho máximo** até o qual o **arquivo pode crescer** (**megabytes** por **padrão**)*
  - *FileGrowth especifica o **incremento** de **crescimento** do **arquivo**. Também é apresentado em **megabytes** por padrão, mas pode ser **especificado** como uma **porcentagem***

SQL Server

- ***Arquivos** e **FileGroups***
  - ***Em vez** de **colocar objetos definidos** pelo **usuário** no **arquivo** de **dados principal**, você tem a **opção** de **adicionar** um **arquivo** de **dados secundário** em seu **BD***
  - ***Em geral**, os **arquivos principais** têm o **sufixo .mdf**, enquanto os **arquivos secundários** têm o **sufixo .ndf***
  - ***Eles não** são **obrigatórios**, todavia, a **prática recomendada** é **utilizar** essas **extensões***

SQL Server

- ***Arquivos** e **FileGroups***
  - *Os **arquivos** de **dados secundários** muitas vezes são utilizados para **propagar dados** pelos **subsistemas** de **disco** ou para **adicionar mais espaço** em **disco** para um **BD**, no caso de **outros arquivos** de **dados** terem **atingido** a **capacidade máxima***
  - ***Outra prática recomendada** é **agrupar** os **arquivos** utilizando **FileGroups***
  - ***Quando** um **BD** é criado, o **FileGroup principal**, que contém o **arquivo** de **dados principal**, é gerado*

- ***Arquivos e FileGroups***

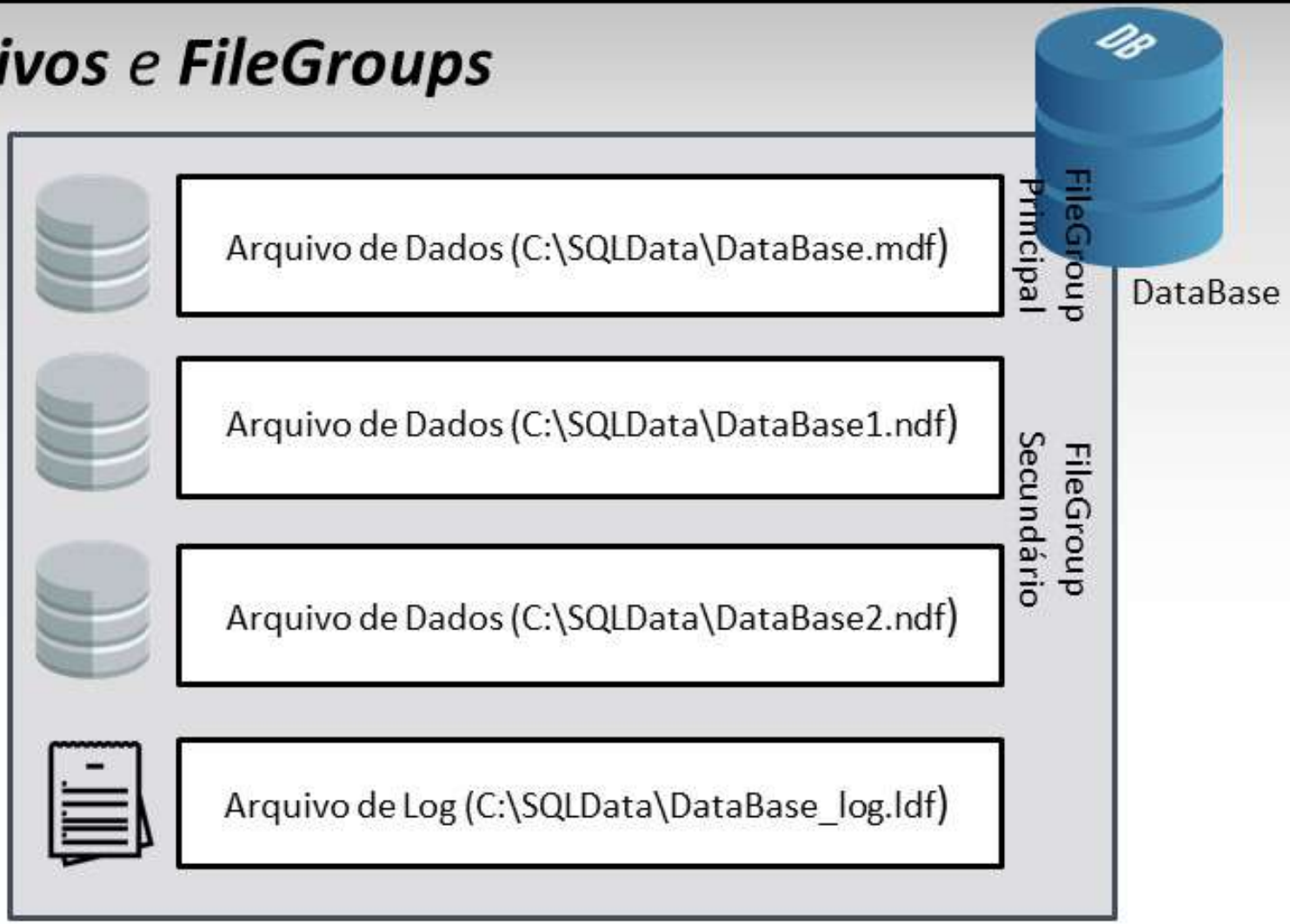


Figura 2 – Arquivos e filegroups do banco de dados

- ***Arquivos** e **FileGroups***
  - ***Exemplo (03): - Parte 1***
    - ***Adicionando** um **arquivo** e um **FileGroups** a um BD*

```
USE master;

ALTER DATABASE bkBankAula10
  ADD FILEGROUP bkBankAula10Group1;
```

- ***Arquivos e FileGroups***
  - ***Exemplo (03): - Parte 2***
    - ***Adicionando** um **arquivo** e um **FileGroups** a um BD*

```
USE master;

ALTER DATABASE bkBankAula10
 ADD FILE
 (
  NAME='bkBankAula10a',
  FILENAME='D:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\bkBankAula10a.ndf',
  SIZE=10MB,
  MAXSIZE=20,
  FILEGROWTH=10%
 ) TO FILEGROUP bkBankAula10Group1;
```

SQL Server

- ***Desanexe** e **Anexe BDs***
  - ***Suponha** que **você queira redistribuir** o **espaço livre** em um **servidor** ou **desativar** um **servidor**, o que **exigiria desanexar** um **BD** de **uma instância** de **SQL Server** e, então, **anexá-lo** a uma **nova instância***
  - ***Existem duas maneiras** de **anexar** e **uma** de **desanexar** um **BD** de uma **instância** de **SQL Server***
  - *Para **anexar** um **BD**, use **sp_attach** ou **CREATE DATABASE**, especificando o **argumento FOR ATTACH***

- ***Desanexe** e **Anexe BDs***
  - ***Exemplo (04):***
    - ***Desanexando** um **BD usando T-SQL***

```
USE master;

EXEC sp_detach_db @dbname = 'DBSample';
```

- ***Desanexe** e **Anexe** BDs*
  - ***Exemplo (05):***
    - ***Anexando** um **BD usando T-SQL***

```
USE master;

CREATE DATABASE DataBaseForSample ON
(FILENAME = 'C:\SQLData\DataBaseForSample1.mdf'),
(FILENAME = 'C:\SQLData\DataBaseForSample2.ndf'),
(FILENAME = 'C:\SQLData\DataBaseForSample_log.ldf')
FOR ATTACH;
```

- ***Desanexe** e **Anexe** BDs*
  - ***Exemplo (06):***
    - ***Anexando** um **BD usando T-SQL (MyAdventureWorks)***

```
USE master;

CREATE DATABASE MyAdventureWorks ON
(FILENAME = 'D:\YourPath\AdventureWorks_Data.mdf')
FOR ATTACH;
```

Restore

- ***Modelos** de **Recuperação***
  - *Um **banco** de **dados SQL Server** pode ser **definido** com um **três modelos** de **recuperação** (o **modelo** **determina** a **precisão** com que um **BD** pode ser **restaurado**)*
    - ***Simple***
    - ***Full***
    - ***Bulk-logged***
  - ***Modelo Simple***
    - ***Não permite backups** de **log** de **transação***
    - ***Não** é **possível restaurar** um **BD** a um **ponto** de **tempo***
    - *O **banco** de **dados** fica **vulnerável** à **perda** de **dados** ao se **usar esse modelo***

Restore

- ***Modelos** de **Recuperação***
  - ***Modelo Full***
    - *A **perda** é **mínima** quando o **backup** do **log** de **transação** é **feito** com **regularidade***
    - ***Cada transação** é **totalmente registrada** no **log** de **transação**, e este **continuará crescendo** até que seu **backup** seja feito*
    - ***Embora** esse **modelo aumente** a **carga administrativa**, seus **dados ficam protegidos contra perda***

Restore

- ***Modelos** de **Recuperação***
  - ***Modelo Bulk-Logged***
    - *As **operações** em **bloco** são **registradas** de **forma mínima**, o que **reduz** o **tamanho** do **log** de **transação***
    - ***Não elimina** a **necessidade** de **fazer backup** do **log** de **transação***
    - *Ao **contrário** do **modelo** de **recuperação full**, no **modelo bulk-logged** é possível **restaurar somente até** o **final** de **qualquer backup***
    - ***Não permite restaurar** até **algum ponto** no **tempo***

TABLES

Tables

- ***Objetivos***
  - ***Desenvolver** um **padrão** de **nomenclatura***
  - ***Entender** os **esquemas***
  - ***Entender** os **diferentes tipos** de **dados** do **SQL Server***
  - ***Entender** as **propriedades** de **coluna***
  - ***Criar** e **alterar tabelas***
  - ***Entender** as **colunas calculadas***
  - ***Adicionar restrições** a **uma tabela***
  - ***Entender** o **recurso FileTable***

Tables

- ***Nomenclatura***
  - *O **primeiro passo** em **qualquer** **projeto** de **design** de **BD** é **criar** um **padrão** de **nomenclatura** a ser usado **durante** o **processo** de **design***
  - ***Embora** a **criação** de um **padrão** de **nomenclatura** **não** seja um **requisito absoluto**, **não** **tê-lo** poderia **gerar** um **banco** de **dados** **desorganizado** que poderia **apresentar** **dificuldades** aos **desenvolvedores** para **acessar** os **dados***
  - ***Padrões** de **nomenclatura** **inconsistentes** em **geral** **inibem** o **processo** de **desenvolvimento** **indiretamente***

Tables

- ***Nomenclatura***
  - ***Para um desenvolvedor*** *que esteja* ***escrevendo código*** *em* ***T-SQL*** *para* ***modificar*** *ou* ***recuperar dados****, os* ***padrões*** *de* ***nomenclatura*** *oferecem caminhos claros para a* ***construção*** *de* ***instruções T-SQL***

Tables

- ***Nomenclatura***
  - ***Padrões** em **Geral:***
    - ***Não** use **espaços dentro** de um **nome** de **objeto** ou de **uma coluna***
    - ***Caracteres** de **sublinhado** são **aceitáveis**, mas saiba que podem **apresentar** alguns **desafios** com **ferramentas** de **visualização***
    - ***Utilize PascalCase**, que significa **colocar** a **primeira letra** de **cada palavra** em **maiúscula** para **nomes** de **objeto** ou **coluna***
    - ***Não** utilize **palavras reservadas***
    - ***Nomes** de **tabela** e **coluna** no **plural** são **aceitáveis** (isso é apenas uma questão de preferência)*

Schemas

- ***Esquemas***
  - ***Embora*** *um* ***banco*** *de* ***dados*** *seja o* ***principal contêiner*** *de* ***todos*** *os* ***objetos****, os* ***esquemas oferecem outro nível*** *de* ***contenção*** *e* ***organização*** *dentro de um* ***banco*** *de* ***dados***
  - ***Utilizando*** *um* ***esquema****, um usuário pode* ***agrupar objetos*** *de* ***escopo*** *ou* ***posse semelhante***
  - ***Por padrão****, o* ***esquema proprietário*** *do* ***banco*** *de* ***dados*** *(****dbo*** *–* ***database owner****) é* ***criado automaticamente dentro*** *de* ***um banco*** *de* ***dados***
  - *Todo* ***objeto criado*** *é* ***adicionado*** *a* ***esse esquema***

Schemas

- ***Esquemas***
  - ***Exemplo (07):***
    - ***Criando o esquema intitulado de Sales e HumanResources***

```
USE bkBankAula10;
GO
CREATE SCHEMA Sales;
GO
CREATE SCHEMA HumanResources;
GO
```

Schemas

- ***Esquemas***
  - ***Exemplo (07):***

Tente **criar todos** os **esquemas antes** de **criar** as **tabelas**. Se isso **não** for **possível**, você sempre pode **mover uma tabela** ou **qualquer outro objeto** de **um** para **outro**, usando a **instrução** ALTER SCHEMA ... TRANSFER.

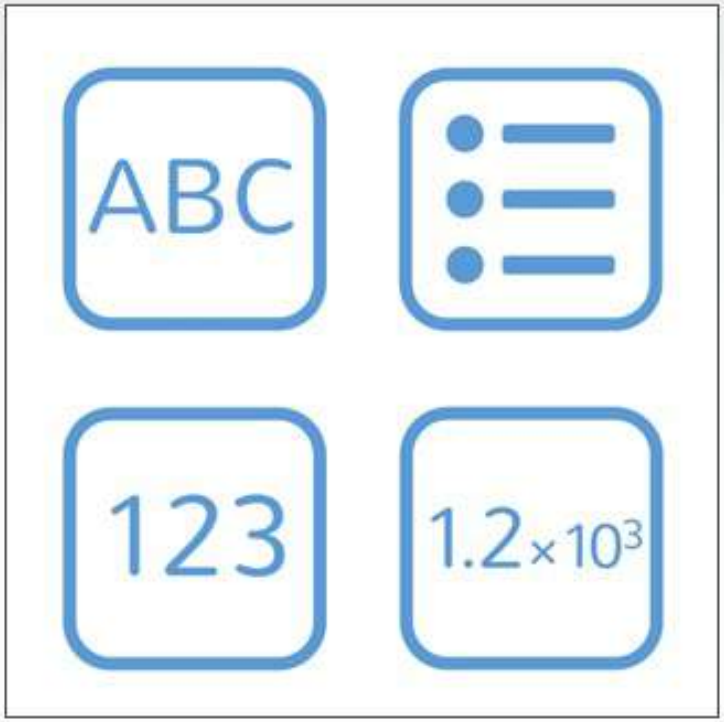
Um **último detalhe** a mencionar sobre os **esquemas** é que podemos **conceder** aos **usuários permissões** para os **esquemas**.

Data Types

- ***Tipos** de **Dados***
  - *O **SQL Server contém quatro categorias distintas** de **tipos de dados: numéricos**, **data** e **hora**, **strings** e **outros***
  - ***Cada uma** das **quatro categorias** contém **subcategorias***
  - ***Todas** as **colunas dentro** de **uma tabela**, **variáveis declaradas** e **parâmetros** devem ter um **tipo** de **dado** correspondente*
  - *Um **tipo** de **dado** **especifica** simplesmente **qual tipo** de **dado** pode ser **colocado** no **objeto** (**coluna**, **variável**, **parâmetro**, etc.)*

Data Types

- ***Tipos** de **Dados***
  - *A **integridade** do **banco** de **dados** depende decisivamente dos **tipos** de **dados** com **escopo apropriado***



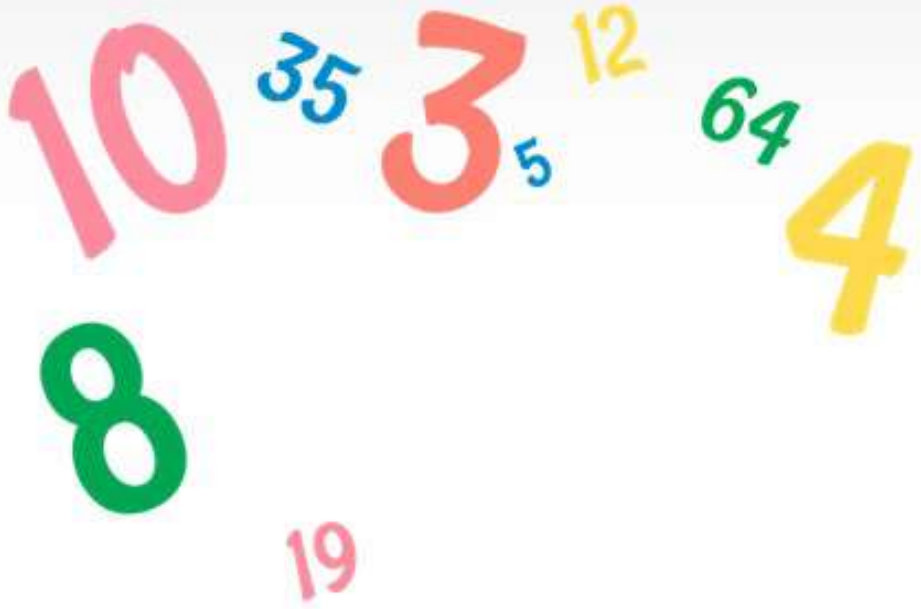
Data Types

- ***Tipos** de **Dados***
  - ***Numéricos***
    - *Os **tipos** de **dados numéricos** têm **duas subcategorias:** **inteiros** e **reais***

Tipo de Dado	Intervalo	Armazenamento
bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	8 bytes
int	-2.147.483.648 a 2.147.483.647	4 bytes
smallint	-32.768 a 32.767	2 bytes
tinyint	0 a 255	1 byte
money	-922.337.203.685.477,5808 a 922.337.203.685.477,5807	8 bytes
smallmoney	-214.748,3648 a 214.748,3647	4 bytes

Data Types

- ***Tipos** de **Dados***
  - ***Numéricos***
    - ***Decimal** e **Numérico: permitem casas decimais**, as quais são **restritas** por **dois valores: precisão** e **escala***



Data Types

- ***Tipos** de **Dados***
  - ***String***
    - ***Contêm três subcategorias: caracter, unicode** e **binário***
    - ***char(n): tipo** de **dado string** de **comprimento fixo** (1 e 8000)*
    - ***varchar(n): tipo** de **dado string** de **comprimento variável** que pode **armazenar** até **2 GB** de **dados***
    - ***text: tipo** de **dado descontinuado** (substituído por um varchar(max))*
    - ***nchar(n): tipo** de **dado string** de **comprimento fixo**, com **comprimento** de **string** entre 1 e 4000*

Data Types

- ***Tipos** de **Dados***
  - ***String***
    - ***nvarchar(n): tipo** de **dado string** de **comprimento variável** que pode **armazenar** até **2 GB** de **dados***
    - ***ntext: tipo** de **dado descontinuado** (substituído por nvarchar(max))*
    - ***binary(n): tipo** de **dado binário** de **comprimento fixo**, com **comprimento** de **string** 1 e 8000*
    - ***varbinary(n): tipo** de **dados binário** de **comprimento variável**, com **comprimento** de **string** de **até 2 GB***
    - ***image: tipos** de **dados descontinuado** (substituído por varbinary)*

Data Types

- ***Tipos** de **Dados***
  - ***String***
    - ***É recomendável** usar **tipos** de **dados** de **comprimento fixo** (**char**, **nchar**, **binary**) em **todas** as **subcategorias**, quando os **valores** que estão sendo **armazenados** tiverem um **tamanho consistente***
    - ***Quando** os **valores não forem consistentes**, podemos utilizar **tipos** de **dados** de **comprimento variável** (**varchar**, **nvarchar**, **varbinary**)*

Data Types

- ***Tipos** de **Dados***
  - ***Data** e **Hora***
    - ***time(n):** armazena a **hora** do **dia sem reconhecer** o **fuso horário** com **base** em um **relógio** de **24 horas***
    - ***date:** armazena um **valor** de **data entre 01-01-01** e **12-31-9999***
    - ***smalldatetime:** armazena um **valor** de **data** e **hora**. O **valor** da **data** é **entre 1/1/1900** e **6/6/2079***
    - ***datetime: similar** a **smalldatetime**, mas oferece um **intervalo** de **data maior** e um **nível** de **precisão** mais **alto** com **relação** ao **tempo***
    - ***datetime2(n): similar** a **datetime**, mas oferece **mais flexibilidade** de **tempo***

Data Types

- ***Tipos** de **Dados***
  - ***Data** e **Hora***
    - ***datetimeoffset:** tipo de dado que **inclui todas** as **características** de **datetime2** e também **reconhece** o **fuso horário**. Torna único entre os **tipos** de **dados** de **data** e **hora**. Permite **armazenar** a **diferença** de **fuso horário** junto com a **data** e **hora***

Data Types

- ***Tipos** de **Dados***
  - ***Outros tipos** de **dados***
    - ***cursor:** uma **cópia temporária** dos **dados** que será usada para **processos recursivos** ou **iterativos** (não pode ser incluído como parte de uma tabela)*
    - ***rowversion(timestamp): gera automaticamente** um **valor** de **8 bytes***
    - ***hierarchyid: representa uma posição** em **uma hierarquia***
    - ***sql_variant: considerado camaleão** entre os **tipos** de **dados**. Pode **assumir** a **identidade** de **praticamente** qualquer **tipo** de **dado** da **lista** de **tipos** de **dados** do **SQL Server***

Data Types

- ***Tipos** de **Dados***
  - ***Outros tipos** de **dados***
    - ***xml: armazena dados XML reais***
    - ***geospatial:** o **SQL Server suporta dois tipos** de **dados geoespaciais: GEOGRAPHY** e **GEOMETRY***
    - ***filestream:** permite o armazenamento de **dados não estruturados**, como **documentos** e **imagens***

Columns

- ***Propriedades** das **Colunas***
  - *A **propriedade mais comum** é **Allow Nulls***
  - ***Permite inserir** uma **linha** na **tabela sem fornecer** um **valor***
  - ***Exemplo:***
    - ***Imagine** que **existe** uma **tabela contendo** **FirstName**, **MiddleName** e **LastName** (nome, nome do meio e sobrenome)*
    - ***Nem todo mundo** tem **um nome** do **meio**, portanto, esse **valor** deve ser **opcional***
    - *Ao **projetar** a **tabela**, considere a **lógica** do **negócio** por trás do valor ao **decidir** sobre sua **nulidade***

Columns

- ***Propriedades** das **Colunas***
  - *A **propriedade mais comum** é **Allow Nulls***
  - ***Permite inserir** uma **linha** na **tabela sem fornecer** um **valor***
  - ***D***

NULL é um valor especial no mundo dos banco de dados. Isso não significa vazio, ao contrário, representa a ausência de um valor e é diferente de uma string vazia

leName

  - ***Nem todo mundo** tem **um nome** do **meio**, portanto, esse **valor** deve ser **opcional***
  - *Ao **projetar** a **tabela**, considere a **lógica** do **negócio** por trás do valor ao **decidir** sobre sua **nulidade***

Columns

- ***Propriedades** das **Colunas***
  - *A **segunda propriedade** mais **comum** é **Is Identity***
  - ***Disponível** para a **maioria** dos **tipos** de **dados numéricos***
  - ***Quando** esse **valor** é **definido** para **uma coluna**, o **SQL Server gera** um **número automaticamente**, à medida que **cada linha** é **inserida***
  - *É possível **personalizar** ou **configurar** o **ponto** de **partida** e como o **número** será **incrementado**, utilizando as **propriedades disponíveis***

Columns

- ***Propriedades** das **Colunas***
  - *O **SQL Server apresenta** um **novo mecanismo** de **geração** de **numeração automática, chamado sequence**, que é um **objeto vinculado** ao **esquema** que **gera uma sequência** de **valores numéricos** com base em **certas opções especificadas durante** a **sua criação***

Tables

- ***Tabelas***
  - ***Exemplo (08):***
    - ***Criando a tabela Address, pertinente ao esquema HumanResources***

```
USE bkBankAula10;

CREATE TABLE HumanResources.Address(
AddressID         INT CONSTRAINT nnAddressAddressID NOT NULL IDENTITY(1,1),
StreetAddress     VARCHAR(125) CONSTRAINT nnAddressStreetAddress NOT NULL,
StreetAddress2    VARCHAR(75) CONSTRAINT nnAddressStreetAddress2 NOT NULL,
City              VARCHAR(100) CONSTRAINT nnAddressCity NOT NULL,
State             CHAR(2) CONSTRAINT nnAddressCity NOT NULL,
EmployeeID        INT CONSTRAINT nnAddressEmployeeID NOT NULL
) ON bkBankAula10Group1;
```

Tables

- ***Tabelas***
  - ***Exemplo (09):***
    - ***Criando a tabela Employee, pertinente ao esquema HumanResources***

```
USE bkBankAula10;

CREATE TABLE HumanResources.Employee(
EmployeeID      INT CONSTRAINT nnEmployeeEmployeeID NOT NULL IDENTITY(1,1),
FirstName       VARCHAR(50) CONSTRAINT nnEmployeeFirstName NOT NULL,
MiddleName      VARCHAR(50) NULL,
LastName        VARCHAR(50) CONSTRAINT nnEmployeeLastName NOT NULL
) ON bkBankAula10Group1;
```

Tables

- ***Tabelas***
  - ***Exemplo (10):***
    - ***Adicionando*** *uma* ***nova coluna****, intitulada de* ***Gender*** *na tabela* ***Employee***

```
USE bkBankAula10;

ALTER TABLE HumanResources.Employee
   ADD Gender CHAR(1) CONSTRAINT nnEmployeeGender NOT NULL;
```

Tables

- ***Colunas Calculadas***
  - ***Além*** *de* ***adicionar dados diretamente*** *em* ***colunas****, é permitido* ***derivar dados*** *de* ***outras colunas***
  - ***Essas colunas*** *são* ***conhecidas*** *como* ***calculadas*** *(aprimoram os dados armazenados nas colunas tradicionais)*
  - ***Exemplo (11):***
    - ***Adicionando*** *uma* ***nova coluna (calculada)****, intitulada de* ***FullName*** *na tabela* ***Employee***

```
USE bkBankAula10;

ALTER TABLE HumanResources.Employee
   ADD FullName AS LastName + ' ' + FirstName;
```

Tables

- ***Restrições***
  - *O **principal objetivo** da **maioria** das **restrições** é a **integridade** dos **dados**, otimizando a **validade** e a **consistência** dos **dados***
  - ***Primary Key***
    - ***É uma coluna** que **contém** uma **lista** de **valores únicos***
    - *Em geral, **uma coluna** de **inteiros** é **adicionada** a **uma tabela** com a **propriedade** de **identidade** e é **utilizada** como **chave primária***
    - ***É possível criar** uma **chave primária** a partir de **praticamente qualquer coluna** ou **combinação** de **colunas***
    - *As principais **limitações** são que a **coluna não pode permitir valores nulos**, o **valor** deve ser **único** e só pode haver **uma chave primária** para elas*

Tables

- ***Restrições***
  - ***Default***
    - ***São perfeitas*** *quanto* ***existe uma coluna*** *que, em geral,* ***contém*** *um* ***valor específico***
    - ***Uma candidata muito boa*** *para isso é* ***uma coluna*** *que tem o* ***tipo*** *de* ***dado bit***
    - *O* ***tipo*** *de* ***dado bit aceita*** *1 ou 0 (****verdadeiro*** *ou* ***falso****)*
    - *Se* ***adicionarmos*** *uma* ***coluna Active*** *à* ***tabela Employee****,* ***especificando*** *se um* ***funcionário*** *está* ***trabalhando*** *na empresa* ***atualmente****, é provável que o* ***valor padrão*** *seja* ***verdadeiro****, ou 1*

Tables

- ***Restrições***
  - ***Unique***
    - ***Frequentemente*** *são* ***confundidas*** *com as de* ***primary key***
    - ***Essas restrições*** *simplesmente* ***garantem*** *que* ***não*** ***possam*** *ser* ***inseridos valores duplicados*** *na* ***coluna correspondente***
    - ***Por exemplo****, vamos supor que precisamos* ***adicionar*** *na* ***tabela Employee*** *uma* ***coluna*** *para* ***números*** *de* ***inscrição*** *da* ***previdência social***
    - *Como esses* ***números*** *são* ***valores realmente únicos****, devemos* ***adicionar*** *uma* ***restrição unique*** *para* ***garantir*** *que* ***determinado número*** *de* ***inscrição*** *apenas seja* ***inserido uma vez***

Tables

- ***Restrições***
  - ***Check***
    - ***Permite*** ***conferir*** *(****verificar****)* *o* ***valor*** *que está* ***sendo inserido*** *em* ***relação*** *a* ***expressões lógicas***
    - *Essa* ***restrição*** *é* ***semelhante*** *à* ***coluna*** *de* ***foreign key****, pois* ***controla*** *os* ***valores inseridos***
    - *A* ***coluna*** *de* ***foreign key*** *recebe* ***valores*** *de* ***outra tabela****, enquanto as* ***restrições check*** *utilizam* ***expressões***

Tables

- ***Restrições***
  - ***Foreign Key***
    - *A **integridade** de **dados** é a **preocupação mais importante** em um **banco** de **dados***
    - *As **chaves estrangeiras** desempenham um **papel vital** na **imposição** da **integridade referencial** do **banco** de **dados***
    - ***Quando configurada** em uma determinada **coluna**, apenas permite a **inserção/alteração** de **valores previamente existentes** na **coluna** a qual é **referenciada***

Tables

- ***Constraints***
  - ***Exemplo (12):***
    - ***Adicionando*** *duas* ***restrições*** *(****NOT NULL****) na tabela* ***Employee***

```
USE bkBankAula10;

ALTER TABLE HumanResources.Employee
  ADD SocialSecurityNumber VARCHAR(10)
        CONSTRAINT nnEmployeeSocialSecurityNumber NOT NULL;

ALTER TABLE HumanResources.Employee
  ADD Active BIT
    CONSTRAINT nnEmployeeActive NOT NULL;
```

Tables

- ***Constraints***
  - ***Exemplo (13):***
    - ***Adicionando restrição de chave primária na tabela Employee***

```
USE bkBankAula10;

ALTER TABLE HumanResources.Employee
  ADD CONSTRAINT pkHumanResourcesEmployeeID
    PRIMARY KEY(EmployeeID);
```

Tables

- ***Constraints***
  - ***Exemplo (14):***
    - ***Adicionando restrição de chave primária na tabela Address***

```
USE bkBankAula10;

ALTER TABLE HumanResources.Address
  ADD CONSTRAINT pkAddressAddressID
    PRIMARY KEY(AddressID);
```

Tables

- ***Constraints***
  - ***Exemplo (15):***
    - ***Adicionando restrição*** *de* ***valor padrão*** *para a* ***coluna Active*** *na tabela* ***Employee***

```
USE bkBankAula10;

ALTER TABLE HumanResources.Employee
  ADD CONSTRAINT dfEmployeeActive
    DEFAULT(1) FOR Active;
```

Tables

- ***Constraints***
  - ***Exemplo (16):***
    - ***Adicionando*** ***restrição*** *de* ***unicidade*** *para a* ***coluna*** ***SocialSecurityNumber*** *na tabela* ***Employee***

```
USE bkBankAula10;

ALTER TABLE HumanResources.Employee
  ADD CONSTRAINT uqEmployeeSocialSecurityNumber
    UNIQUE(SocialSecurityNumber);
```

Tables

- ***Constraints***
  - ***Exemplo (17):***
    - ***Adicionando restrição*** *de* ***chave estrangeira*** *para a* ***coluna*** ***EmployeeID*** *da tabela* ***Address***

```
USE bkBankAula10;

ALTER TABLE HumanResources.Address
  ADD CONSTRAINT fkAddressEmployeeID
    FOREIGN KEY(EmployeeID)
          REFERENCES HumanResources.Employee(EmployeeID);
```

Tables

- ***FileTable***
  - ***Complementa*** *a* ***tecnologia*** ***FileStream*** *existente (é preciso de habilitar os recursos de FileStream antes de criar uma FileTable)*
  - *O* ***recurso*** *de* ***FileTable*** *permite* ***armazenar vários tipos*** *de* ***documentos****, permitindo* ***consultar diretamente*** *os* ***atributos expostos*** *pelo* ***sistema*** *de* ***arquivos*** *do* ***Windows*** *usando* ***T-SQL***
  - ***FileStream*** *é um* ***recurso avançado*** *do* ***SQL Server***
  - ***Serviços integrados*** *do* ***SQL Server****, como* ***pesquisas full-text*** *e* ***semântica****, podem consultar os dados* ***não estruturados armazenados*** *na* ***FileTable***

EXERCÍCIOS

Referências

Noble, E.; Pro T-SQL 2019 Toward Speed, Scalability, and Standardization for SQL Server Developers. Apress, 2020.

Ben-Gan, I.; Microsoft SQL Server 2012 T-SQL Fundamentals. Pearson Education. 2012.

Lahoud, P.; Lopes, P.; T-SQL Querying: A guide to developing efficient and elegant T-SQL code. Packt Publishing. 2019.